Interacción Humano Computadora
Proyecto

Carlos Alexis Barrios Bello
zS23000636@estudiantes.uv.mx

Maestría en Inteligencia Artificial

IIIA Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección 2a, No 112
Nuevo Xalapa, Xalapa, Ver., México 91097

19 de junio de 2024

- Variable de entrada: En este sistema de lógica difusa, la variable de entrada es la distancia entre el jugador y el bloque.

```
float distance = Vector3.Distance(player.position, block.position);
```

- Fuzzificación: La fuzzificación es el proceso de convertir un valor nítido (crisp value) en un valor difuso utilizando funciones de pertenencia. En mi caso, se utilizaron tres conjuntos difusos para la distancia: Cerca, Medio y Lejos.

```
float close =
    Mathf.Max(0, Mathf.Min(1, (distanceMedium - distance)
    / (distanceMedium - distanceClose)));
float medium =
    Mathf.Max(0, Mathf.Min((distance - distanceClose)
    / (distanceMedium - distanceClose),(distanceFar - distance)
    / (distanceFar - distanceMedium)));
float far =
    Mathf.Max(0, Mathf.Min(1, (distance - distanceMedium)
    / (distanceFar - distanceMedium)));
```

- Funciones de Pertenencia: Las funciones de pertenencia definen cómo los valores de entrada son mapeados a conjuntos difusos. En este caso, son funciones de pertenencia triangulares.

    1. Cerca: La función de pertenencia es alta cuando la distancia es baja y disminuye linealmente a medida que la distancia aumenta desde distanceClose a distanceMedium.
    2. Medio: La función de pertenencia es alta cuando la distancia está en el medio, disminuyendo linealmente hacia los extremos.
    3. Lejos: La función de pertenencia es alta cuando la distancia es alta y disminuye linealmente a medida que la distancia disminuye desde distanceMedium a distanceFar.

- Reglas de inferencia: Las reglas de inferencia difusa son las que determinan la salida basada en las entradas difusas. Las reglas en este sistema son las siguientes

    1. Si la Distancia es Cerca, entonces la Velocidad es Lenta.
    2. Si la Distancia es Medio, entonces la Velocidad es Media.
    3. Si la Distancia es Lejos, entonces la Velocidad es Rápida.

    Estas reglas se implementan en la función CalculateFuzzySpeed.

- Desfuzzificación: a desfuzzificación es el proceso de convertir los valores difusos de salida en un valor nítido (crisp value). En este caso, se utilizó una media ponderada para combinar las velocidades lenta, media y rápida en un valor único.

```
float slowSpeed = close * minSpeed;
float mediumSpeed = medium * (minSpeed + maxSpeed) / 2;
float fastSpeed = far * maxSpeed;

float totalWeight = close + medium + far;
return (slowSpeed + mediumSpeed + fastSpeed) / totalWeight;
```

    1. Velocidad Lenta (slowSpeed): ponderada por el grado de pertenencia del conjunto Cerca.
    2. Velocidad Media (mediumSpeed): ponderada por el grado de pertenencia del conjunto Medio.

3. Velocidad Rápida (fastSpeed): ponderada por el grado de pertenencia del conjunto Lejos.

El resultado es una velocidad nítida que se utiliza para mover el bloque.

- Valores de Salida: El valor de salida del sistema difuso es la velocidad con la que el bloque se moverá. Este valor se calcula y se utiliza para mover el bloque de manera fluida

```
void MoveBlock()
{
    block.position = Vector3.Lerp(block.position,
        targetPosition, speed * Time.deltaTime);
}
```

Ahora se procede a ver el diagrama de las funciones de membresía:

```
import matplotlib.pyplot as plt
import numpy as np

# Definición de las funciones de membresía
def triangular(x, a, b, c):
    return max(0, min((x-a)/(b-a), (c-x)/(c-b)))

# Valores de entrada
x = np.linspace(0, 10, 500)

# Funciones de membresía
close = [triangular(val, 0, 0, 5) for val in x]
medium = [triangular(val, 0, 5, 10) for val in x]
far = [triangular(val, 5, 10, 10) for val in x]

# Graficar las funciones de membresía
plt.figure(figsize=(10, 6))
plt.plot(x, close, label='Cerca', color='blue')
plt.plot(x, medium, label='Medio', color='green')
plt.plot(x, far, label='Lejos', color='red')

plt.title('Funciones de Membresía')
plt.xlabel('Distancia')
```

```
plt.ylabel('Grado de Membresía')
plt.legend()
plt.grid(True)
plt.show()
```

Junto con su diagrama:

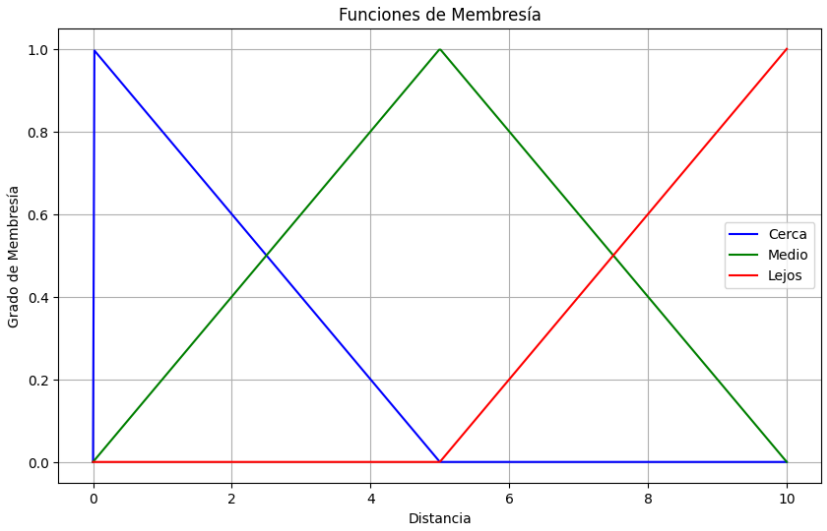

Figura 1: Gráfica de las funciones de membresía

Referencias