

Árboles de decisión

Un árbol de decisión es una estructura que representa de forma jerárquica todas las posibles decisiones que puede tomar un agente (por ejemplo, un personaje controlado por IA) y las consecuencias de cada una. Cada nodo del árbol corresponde a una condición o pregunta; sus ramas son las respuestas posibles y los nodos hoja indican la acción que se ejecutará.

Ejemplo:

```
¿Ve al jugador?  
├── Sí → ¿Está cerca?  
│   ├── Sí → Huir  
│   └── No → Seguir  
└── No → Patrullar
```

¿Cómo se modela?

En lugar de usar if-then-else directamente, en el programa modelamos el árbol como un conjunto de objetos que representan nodos y hojas. Esto permite:

- Modificar el árbol sin tocar el código, incluso usando herramientas gráficas sin necesidad de saber programar.
- Reutilizar la lógica para diferentes NPCs.
- Visualizar y exportar el árbol como imagen o código.

Componentes del árbol

Clase NodoDecision

Evalúa una **condición** y decide si ir por la rama Sí o No.

Está formada por:

- **condicion**: una **función** que devuelve True o False.
- **true_case**: a dónde ir si la condición es True.
- **false_case**: a dónde ir si es False.

Clase Hoja

Representa una acción final que se ejecuta.

Está formada por:

- **accion**: una **función** que modifica el comportamiento del NPC.

¿Cómo se construye un árbol?

Imaginemos que queremos que un NPC haga esto:

¿Ve al jugador?

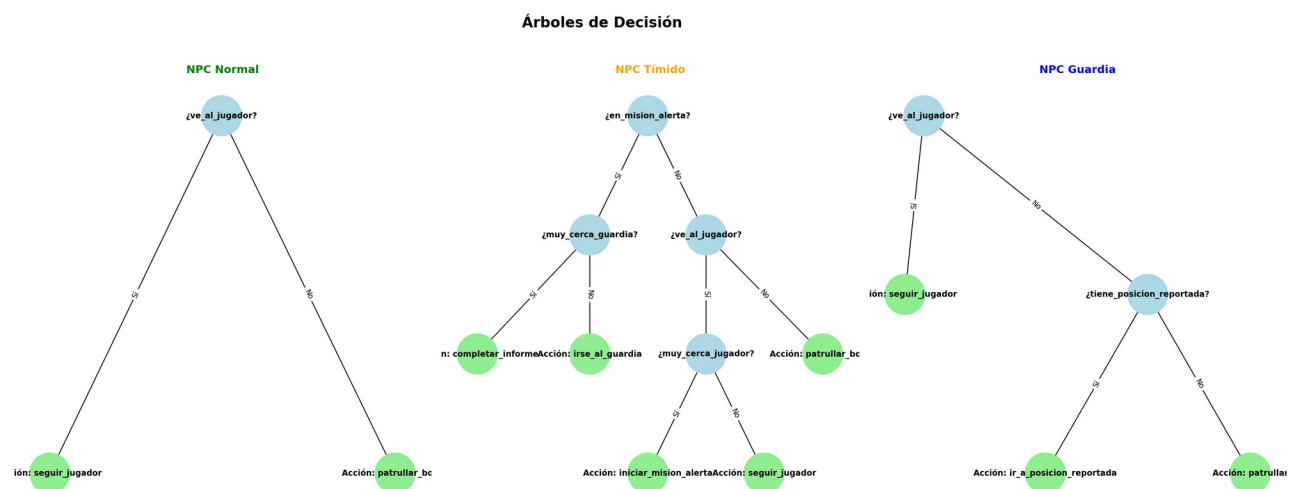
- └ Sí → Seguir al jugador
- └ No → Patrullar borde

Se construye así:

```
# Creamos las acciones
accion_seguir = Hoja(seguir_jugador) # seguir_jugador es una
función python que mueve el NPC hacia el jugador
accion_pasear = Hoja(patrullar_borde) # patrullar_borde es una
función python que mueve el NPC por los bordes del escenario

# Creamos el nodo raíz
nodo_raiz = NodoDecision(ve_al_jugador)
nodo_raiz.true_case = accion_seguir
nodo_raiz.false_case = accion_pasear
```

Ejemplo



En esta imagen se muestra el comportamiento de tres NPCs distintos modelados usando árboles de decisión.

Tracedump:

newBaseSize: 12pt

newBaseSizeInPt: 12