Proyecto

X-Kating

Grupo

WASTED HORCHATA

Documento de diseño técnico del motor de IA

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. Descripción general del motor

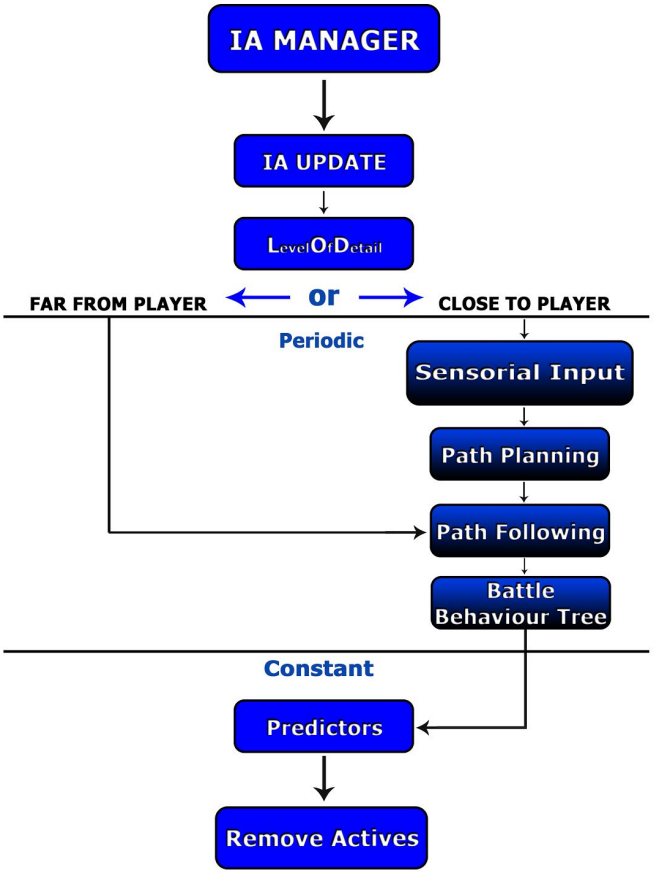
El **motor de IA** se encargará de mover y definir el comportamiento de una serie de adversarios para el jugador. Debe de intentar cumplir los siguientes objetos:

- Los adversarios deberán de **responder** a los estímulos externos **de una manera realista**
- El motor debe ocupar **el menor tiempo de CPU posible.** Como primera meta al menos el **5% del total.**
- Su objetivo es plantear un **enfrentamiento justo pero** que suponga un **reto** al jugador.

Para conseguir eso haremos uso de **una arquitectura** que tenga **categorías prioritarias** sobre tareas, además de algunas técnicas de **balanceo de carga.**

El elemento principal para la gestión del tiempo de ejecución de la IA será el **gestor de recursos**, que **asignará tiempo de proceso** a cada elemento. Con ello consigue **dividir** el **tiempo** de ejecución de **forma equitativa** entre cada iteración y **evita** grandes **sobrecargas** en puntos donde deban tomarse varias decisiones de golpe.

2. Estructura general del motor



3. Descripción del diagrama

El **IA Manager** se encarga de gestionar el **funcionamiento del sistema**, y es el que da paso a cada Update de la IA. Cada **Update** de la IA lleva la información relativa del gestor de recursos y se **gestiona** la cantidad de **elementos** que **deben actualizarse, el tiempo de proceso de cada uno y** en qué **orden**.

Luego el **Level of Detail** (**LoD**), dependiendo del tiempo de ejecución y otras variables, **decide el nivel de complejidad** que se aplicará al update de la IA **de cada NPC**. La **principal decisión** será **en base** a una **distancia** prefijada **relativa al player**, luego el nivel de complejidad en el cálculo de las rutas de carrera y de las variables de conducción.

Luego se **ejecuta** el **comportamiento de la IA. Según** el **ciclo** en el que nos encontremos, se ejecutan funciones **periódicas y constantes.**

Las funciones **periódicas** se repiten cada tantos updates o tiempos asignados. Consisten principalmente en:

1. **Sensorial inputs:** Los NPCs obtendrán información de:
    a. Raycasts que emanen de ellos.
    b. Su cono de visión.
    c. Oído por rango.
    d. Variables locales de posición y velocidad.
2. **Path planning:** combina la información previamente obtenida con el sistema de **Lógica Difusa y** el de **WayPoints.** Buscamos los siguientes WayPoints de la pista y la posible presencia de obstáculos en el camino para determinar hacia donde debe de ir el NPC. Debe de cumpplir que:
    a. **Tener en cuenta elementos ventajosos** como cajas, físicas del terreno o rampas aceleradoras.
    b. **Definimos el vector dirección** del NPC en función del WayPoint que tiene asignado. El punto seleccionado en el WayPoint será **más lejano cuanto mayor sea la velocidad** del NPC.
3. **Path following:** Se divide en **dos fases:**
    a. **Collision Avoidance:** el NPC **valorará** si el punto escogido a seguir produce una **colisión en su ruta**. En caso de que así sea, **decrementará su velocidad para evitarlo** y escogerá un punto en el WayPoint más cercano en el siguiente update. **Girará** en caso de que sea necesario.
    b. El vehículo **se moverá** siguiendo su vector de dirección **usando lógica difusa**. Devolverá una acción recogida por el **input manager**, que lo hará acelerar, frenar, o girar en algún sentido. Los resultados serán los mismos que si **el NPC usará Joystick**.
4. **Battle Behaviour Tree:** El NPC lanza el objeto que **tenga en el inevntario.** Usa un **árbol de comportamiento** que determine **cuándo** y cómo lanzar el objeto de tal manera **que parezca realista al jugador**. **Este árbol recibirá:**
    a. La posición actual del player
    b. La posición cercana de los demás jugadores

c. Sus posiciones en el marcador
d. Variables sensoriales pertinentes.

El resultado debería de ser un lanzamiento que reflejase una ventaja para el NPC frente a los demás jugadores, siendo **más óptimo cuanto más difícil** sea el nivel de la IA.

Las funciones **constantes** deben de realizarse en cada update. **Consisten** en: **tareas compensatorias y la etapa de exclusión.**

Las **tareas compensatorias**:

1. **Actualizan el comportamiento** de la IA que se ha quedado sin órdenes este frame.
2. **Realizan cálculos sencillos**
3. Decididas por el Resource Manager o por la periodicidad de otras funciones.
4. **Casos de uso:**
    a. **Interpolación lineal** del movimiento de un NPC.
    b. **Evitar los cálculos** del “Sistema de batalla” cuando un NPC no ha pasado por encima de una caja de objetos.
    c. **Evitar cálculos** de Path planning o following cuando un jugador ha sido noqueado (aunque dure un segundo o dos).

La **etapa de exclusión** quita los componentes a actualizar en el update. **Casos de uso**:

1. **Objeto** es **destruido** en el juego
2. **Objeto** es **desactivado** durante X tiempo.
3. **Objeto** es **paralizado**.
4. **Objeto** con el **componente** de **IA desactivado** o destruido por un determinado proceso.