

La IA que modifica la dificultad de un juego en tiempo real

DOCUMENTO DE AYUDA PARA LA IDEA DEL PROYECTO

CARLOS CASTILLO GONZÁLEZ

Índice de contenidos

1. Resumen de la idea	2
2. Especificaciones del videojuego.....	3
2.1. Tipo de juego	3
2.2. Personaje Principal.....	3
2.3. Enemigos	3
2.4. Puntuación.....	3
3. Problema: entrenamiento de la IA.....	4
3.1. Inputs.....	4
3.2. Outputs.....	4
3.3. Determinación del nivel del jugador	4
3.4. Conjunto de entrenamiento, como entrenar la IA	4
4. Tecnología	5
5. Modelos de IA a usar.....	6
6. Otros temas	7

1. Resumen de la idea

En los videojuegos el nivel de dificultad se define de varias formas:

1. Selección por parte del jugador de la dificultad: Muy fácil, fácil, normal, difícil y muy difícil. El problema es que jugador podría encontrarse un reto demasiado complicado, o algo demasiado fácil que fuera aburrido.
2. Sistema de niveles: dividir el mapa en zonas y a cada zona un nivel, si quieres ir a una zona con un nivel mucho más alto no puedes, ya que morirías instantáneamente.
3. Sistema de niveles el cual a medida que vas pasando pantallas los niveles son más complicados, puede llegar un nivel que el jugador no sea capaz de pasarse.
4. Sistema de oleadas de enemigos: ejemplo call of duty zombies, a medida que pasan oleadas vienen más enemigos.

A demás de esto la inteligencia Artificial en videojuegos algunos no la consideran como inteligente y normalmente se utilizan máquinas de estado, algoritmos como A* o incluso árboles de decisión para ellas, (Limitaciones de los videojuegos entre físicas, IA y gráficos).

Con todo esto propongo usar algún algoritmo de la rama de inteligencia artificial subsimbólica, es decir redes neuronales y modelos de aprendizaje automático que sea capaz de modificar la dificultad del juego mientras el jugador juega para que el propio juego “acompañe” al jugador en base a sus capacidades en el juego.

Crearía el juego máximo 1 semana de trabajo, dedicaría el resto del tiempo al entrenamiento de la IA para que sea capaz de generalizar el problema para cualquier jugador.

2. Especificaciones del videojuego

2.1. Tipo de juego

- Plataformas 2D:
- Escenario Finito
 - o Dividirse en niveles, y niveles adaptados al jugador; ejemplo 5 niveles y dependiendo de como de bueno seas el siguiente será más fácil o difícil.
- Escenario procedural
 - o Escenario continuado y sin fin pensado para alcanzar una puntuación en base al tiempo de juego, distancia o enemigos eliminados o un conjunto de estos.

2.2. Personaje Principal

- Movimiento: Salto, Izquierda o Derecha**.
- Ataque ejemplo Salto + Derecha o como el Mario caerles encima**.
- Vidas, Ejemplo 10 y si obstáculo o enemigo chocan, entonces -1**.

2.3. Enemigos

- Obstáculos
- NPC: Entonces pueden ser eliminados, chocar o disparar, IA Simbólica simple para estos**.

2.4. Puntuación

- Si procedural: Distancia recorrida + enemigos eliminados** (Vida sin sentido), 1 toque = muerte.
- Si finito: Tiempo en pasarse un nivel + Enemigos eliminados del nivel** + ¿Vidas restantes (o – vidas perdidas)**.

3. Problema: entrenamiento de la IA

3.1. Inputs

- Procedural:
 - o Puntuación: Más alta juegas mejor, más baja juegas peor.
 - o Enemigos eliminados: Más eliminados mejor, menos peor**.
- Finito:
 - o Tiempo en pasarse un nivel: Menor mejor, mayor peor.
 - o Enemigos eliminados: Mayor numero mejor, menor peor**.
 - o Vida: Más vidas restantes mejor, menos vidas restantes peor (Más vidas perdidas peor, menos vidas perdidas mejor) **.

3.2. Outputs

- Aumento o disminución de la dificultad:
 - o Procedural: Ir más rápido o más lento, aparición más o menos frecuente de enemigos u obstáculos**.
 - o Finito: Niveles más o menos complejos que los anteriores, Complejidad para pasar los obstáculos, saltar, eliminar o esquivar enemigos**.

3.3. Determinación del nivel del jugador

- Procedural: Puntuación, enemigos eliminados**.
- Finito: Velocidad para pasarse el nivel, Enemigos eliminados**, Vidas restantes o perdidas**

3.4. Conjunto de entrenamiento, como entrenar la IA

- Diferentes jugadores
 - o Jugador: Diferentes IAs capaces de reconocer el terreno y actuar en consecuencia. Interesante: Aprendizaje reforzado** (Mejor opción)
 - o Subirlo a una plataforma y que muchos jugadores jueguen mucho.

4. Tecnología

- Godot: Blueprints (Drag and drop) y GDScript parecido a Python (consideraría tiempo de aprendizaje del engine) (Posible mejor opción).
- Unity: C#, “menor” tiempo de aprendizaje del engine, Paquete de ML propio de Unity.
- Game maker studio 2: Blueprints (Drag and drop) y C++ (Peor opción).
- La IA se haría con los scripts dentro del propio juego.

5. Modelos de IA a usar

- Enemigos: Maquinas de estados o arboles de decisión movimiento de pathfinding.
- Jugador:
 - Máquina de estados
 - Arboles de decisión
 - Sin ninguna de las dos (No sería IA como tal)
 - Interesante: Aprendizaje por refuerzo.
- IA controladora del nivel de dificultad:
 - Unity Machine Learning Agents: El agente sería el propio mundo el cual aumenta o disminuye el nivel de dificultad en base a la puntuación, enemigos eliminados.
 - El nivel de dificultad aumentaría o disminuiría en base a sacar más o menos enemigos u obstáculos. Ejemplo si Out = 0 -> menos enemigos o menos velocidad, si Out = 1 -> Más velocidad, si Out = 2 -> Más velocidad y enemigos.
 - Entrenamiento de varias instancias al mismo tiempo + juego propio.
 - ¿Podría con entrenamiento por refuerzo? -> Que la IA aprenda a jugar al videojuego o que aprenda a como juega el jugador entonces si muy fácil o difícil -> negativo; si balanceado -> positivo; con esto se tendría en cuenta que el jugador va mejorando a medida que juega.

6. Otros temas

- Juego estilo Jump King: juego de plataformas vertical donde la dificultad está en el propio salto. Salto largo y salto corto, las plataformas serían el “enemigo” si eres muy bueno el salto tiene que ser más preciso, es decir, plataformas más pequeñas, y si se te da muy mal plataformas más grandes, no se agrandarían al momento si no al crearse la instancia.
- Alien Isolation: La IA del alien se adapta a como juegas para que no explotes un recurso que haga que el juego sea muy fácil, es decir, si encuentras un arma con mucho daño al ir a repararla el alien no te dejara y no podrás volver a usarla. Son dos IAs una que controla todo el mapa del juego y otra compuesta de 36 ramificaciones de un árbol de decisión.
- Creación de una RN para la IA de un videojuego: Con el paquete de ML de Unity crea una RN como contrincante.
- IA usada para comprobar si las mecánicas de un juego están balanceadas: TFG de 06/2022.