

Universidad Peruana de Ciencias Aplicadas

UPC



Trabajo Hito 1

CURSO DE Inteligencia Artificial

Carrera de Ciencias de la Computación

Sección CC61

Alumno:

Renzo Perez

Rogger Avalos

Andres Pautrat

Monterrico, septiembre 2019

Abstract--Los videojuegos siempre han tenido como reto, que el jugador se sienta desafiado por los enemigos del mismo, sin que este se sienta demasiado injusto, por lo que los enemigos no deben sentirse omniscientes de los movimientos del jugador ni simplemente hacer acciones aleatorias. En este documento se presenta un software capaz de aprender de los enemigos a los que se enfrente (jugadores) mediante un algoritmo genético. Donde se van probando diferentes respuestas a las acciones del jugador oponente en diferentes patrones, de los cuales al final de la pelea se pondrá una puntuación que servirá para escoger qué patrones formarán la siguiente generación.

I. INTRODUCCIÓN

Los videojuegos en la actualidad se han popularizado a nivel global, es difícil encontrar algún joven que nunca haya probado jugar alguno, debido a que ahora son accesibles para casi cualquier persona y es un medio con el cual entretenerse. En una gran cantidad de juegos no tienen una interacción con otro ser humano, por lo que optan por la opción de crear simulaciones que permitan al jugador sentir un reto, una inteligencia o destreza que superar para que el juego pueda ser divertido. Muchas personas entienden esto como necesario y una parte natural de cualquier título al que le dediquen su tiempo.

En este paper se implementará un bot que pueda presentar un reto divertido en un juego de pelea 1 contra 1 en el cual tanto el personaje controlado por el jugador como el personaje controlado por la maquina tendrán un set de movimientos y acciones que deberán saber cómo y cuándo usar para poder derrotar a su oponente. Sin embargo, para que el usuario pueda tener una experiencia placentera la entidad a la que intente derrotar no debe ser tan difícil como para frustrar, ni tan fácil como para aburrirlo. Por lo que se busca crear un bot con un nivel de dificultad promedio que pueda enfrentarse a los jugadores.

Para lograr el objetivo se utiliza un algoritmo genético mejorado con reproducción y mutaciones, el bot tendrá una cantidad de conjuntos de probabilidades igual a la cantidad de movimientos posibles, cada conjunto establecerá cuán probable es que el bot realice una acción en respuesta a otra acción del oponente (jugador), de este modo nuestro bot puede lanzar una respuesta incorrecta para que el usuario no

sienta una injusticia cuando el bot ya haya pasado por una cantidad considerable de generaciones. Para el entrenamiento del mismo se plantea que se realicen con personas reales que prueben el juego e intenten vencer al bot para que este vaya mejorando con la práctica. Al comienzo el bot simplemente tendrá varias combinaciones de valores random para poder aplicar el algoritmo genético y se espera que con el entrenamiento puede volverse un enemigo entretenido en el juego.

II. ESTADO DEL ARTE

A. Trabajos Relacionados

Existen varios papers que han manejado el tema de inteligencia artificial dentro de los videojuegos desde diferentes puntos, de vista tomando diferentes tipos de juegos para la inclusión de estas.

Un problema muy habitual es el manejo de diversos entes individuales que tienen un objetivo en conjunto, como por ejemplo el de capturar a una bandera o derrotar a un enemigo. Según Hu, Z., Zhu, M., Chen, P., y Liu, P la mejor forma de solucionar estos problemas es la evaluación de los individuos de manera singular y la generación de árboles de generación de patrones para los individuos usando cadenas de markov [1]. Bedia, M. G., Castillo, L., Lopez, C., Seron, F., & Isaza, G. proponen una solución diferente para el mismo problema, ellos proponen usando el dilema de exploración explotación, que cada individuo tenga un estado de búsqueda de opciones y un estado de agresión, donde intenta ir directamente al objetivo, con esto se programan X cantidad de bots, y se aplica algoritmo genético seleccionando a los de mejor rendimiento [3]. Por último, para este mismo problema N. L. y M. Martínez proponen la toma de decisiones a través de un árbol de decisiones y máquina de estado finito[4].

Así también, cuando se habla del dilema de exploración explotación es necesario analizar de cada uno en el juego con respecto a su propio perfil. Según Ching-Ying, C., Ya-Hung, C., & Tsung-Che Chiang, C., se pudo llegar a emplear un sistema de dificultad adaptable a cada jugador en el que la dificultad, usando su ejemplo, el mapa de “mario” varía de acuerdo al perfil del jugador [2].

Por otro lado, hablando de la toma de decisiones de la inteligencia artificial del juego, muchos de los autores hacen uso de algoritmos previamente definidos. Un ejemplo de esto se hace presente en el paper de G. Vladimir, B. Endre, E. Khaled y M. Kazuhisa, en el que usan un algoritmo pseudo polinomial para determinar posiciones aleatorias. Esto puede ser usado para que un bot tome decisiones a los posibles movimientos de un jugador. [5]. Pirovano, M., & Lanzi, P. L. propone la utilización de lógica difusa para que con un conjunto de reglas previamente definidas, un personaje podría optar por moverse atacar o huir, cubrirse [8]. Así también, Antonio, G. P. menciona la posibilidad de usar algoritmos genéticos con el algoritmo Ant Colony Optimization para resolver el problema de encontrar un camino del juego “The Lemmings Game”. Ambos algoritmos trabajan sobre una población y poseen una función fit, la cual en nuestro caso podría contrastar las generaciones de luchadores [10].

Cuando se crea un bot es muy importante considerar algún paradigma ético que respetar para que el bot no realice ciertas acciones que podrían llegar a perjudicar a alguien de alguna manera. Para esto P. Anunpattana, H. Iida y C. Panumate ha establecido ciertos enfoques dentro de un paper para servir como guías a quien desee desarrollar un bot. Estos enfoques son:

1. Puede tener innumerables leyes que controlan de sus acciones.
2. Puede calcular las consecuencias de sus acciones y evaluar esas consecuencias acuerdo a principios utilitarios.
3. Puede diseñar con un sistema de aprendizaje con la capacidad de adquirir un comportamiento ético a partir de experiencias(Un sistema de virtudes éticas).

Además, brinda una serie de pasos que una IA debe seguir para poder evaluar de manera correcta qué acciones son apropiadas y justas en la situación específica en la que se encuentre [7].

B. Estudio Autodirigido

- Procesamiento de lenguaje natural: es un campo de las ciencias de la computación, Inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano, el cual

se ocupa de formular e investigar los mecanismos computacionales para la comunicación por el medio de lenguas del mundo(lenguaje natural). PLN trata de diseñar mecanismos para que la comunicación sea eficaz computacionalmente. Los componentes de PLN son:

- Análisis morfológico: Consiste en el análisis interno de las palabras que forman oraciones.
- Análisis sintáctico: Consiste en el análisis de la estructura de las oraciones dependiendo del modelo gramatical empleado.
- Análisis semántico: Identifica la interpretación de las oraciones.
- Análisis pragmático: Incorpora el análisis del contexto de las oraciones.

III. APORTE

El diseño en la cual nos estamos basando para la simulación de un juego de pelea contra bots es el algoritmo genético debido a que en la reproducción y mutación, puede mejorar las siguientes generaciones de luchadores. Cada generación tendrá 5 ejemplares los cuales tendrán batallas individuales contra un jugador y al final de su pelea serán evaluados cada uno de los 5 de manera individual, de estos se escogen los 2 mejores para reproducirse y reemplazará los 2 peores luchadores. Por último, se escogerá de manera aleatoria si a alguno de los 5 patrones de luchadores es modificado con cambios pequeños al azar o si no se cambian. De esta manera cada generación va mejorando según los patrones de movimientos que tuvieron mejores resultado anteriormente, y a veces haciendo modificaciones random con la mutación.

Para el diseño del patrón de cada personaje, primero debe saberse todos los movimientos que se puede realizar, el juego contempla los siguientes movimientos:

Punch: Un golpe a la parte superior del cuerpo.

Kick: Un golpe a la parte inferior del cuerpo.

Adelante: Acerca al personaje al jugador oponente.

Atrás: Aleja al jugador del oponente.

- Fecha izquierda: Movimiento a la izquierda. Se disminuye la posición en el eje x en el juego

- Flecha derecha: Movimiento a la derecha. Se aumenta la posición en el eje x en el juego.
- Espacio: Ataque de patada. Aumenta las magnitudes del hitbox inferior del atacante.
- Enter: Ataque de puños. Aumenta las magnitudes del hitbox superior del atacante.
- Agacharse: Se agacha. El peleador disminuye el tamaño de su hitbox agachándose.
- Pararse: Se pone de pie. El peleador restablece el tamaño normal de su hitbox poniéndose de pie.

OUTPUT:

Para cada uno de los inputs mencionados se definen los siguientes outputs

- Animación a la izquierda
- Animación a la derecha
- Animación “Kick”
- Animación “Punch”
- Animación “Crouch”
- Animación “Get Up”

Comparativa:

El fantástico trabajo de Pirovano, M., & Lanzi, P. L. (2014) influyó en la toma de decisiones de nuestro proyecto[8]. Si bien es cierto Pirovano utilizó la lógica difusa y junto a una gran cantidad de reglas pudo definir las acciones que tomaría su IA en su juego. Sin embargo, nosotros quisimos ir más allá y no limitarnos en una cantidad de reglas o cantidad invariable de movimientos. Cada vez que se termine una ronda de peleas, el jugador se encontrará con un reto mayor. Y así como Bedia, M. G., Castillo, L., Lopez, C., Seron, F., & Isaza, G. proponen de cierta forma dar valor a ese concepto de “exploración explotación”[9] usando el algoritmo genético. Fighter hará uso de ese concepto para representar ese extra que es un inicio para la dificultad dinámica.

VI. CONCLUSIÓN

En resumen, la utilización de algoritmos genéticos en el ámbito de creación de inteligencia artificial en videojuegos aporta una sensación de dificultad progresiva debido a que los bots mejoran cada vez más y el usuario siempre sentirá el reto en el juego. Se piensa implementar una campaña para la

próxima actualización, realizar las animaciones con sprites y hacer el juego de paga. Será viable la compatibilidad con mandos.

REFERENCIAS

- [1] Hu, Z., Zhu, M., Chen, P., & Liu, P. (2019). On convergence rates of game theoretic reinforcement learning algorithms. pg. 90–101.
- [2] Ching-Ying, C., Ya-Hung, C., & Tsung-Che Chiang, C. (2013). *Intelligent Level Generation for Super Mario using Interactive Evolutionary Computation*.
- [3] Bedia, M. G., Castillo, L., Lopez, C., Seron, F., & Isaza, G. (2016). *Designing virtual bots for optimizing strategy-game groups*. *Neurocomputing*, 172, 453–458.
- [4] N. L. y M. Martínez, «STACRAFT: a Starcraft-based educational platform for Artificial».
- [5] G. Vladimir, B. Endre, E. Khaled y M. Kazuhisa, «A pseudo-polynomial algorithm for mean payoff stochastic games with perfect information and few random positions,» 2019.
- [6] P. Anunpattana, H. Iida y C. Panumate, «Finding Comfortable Settings of Snake Game Using Game Refinement,» Japan, 2016.
- [7] F. Alaieri y A. Vellino, «A Decision Making Model for Ethical (Ro)bots,» Canada , 2017.
- [8] Pirovano, M., & Lanzi, P. L. (2014). *Fuzzy Tactics: A scripting game that leverages fuzzy logic as an engaging*. Milano, Italia: Elsevier Ltd.
- [9] Bedia, M., Castillo, L., Lopez, F. S., & Isaza, G. (2015). *Designing virtual bots for optimizing strategy-game*. *Neurocomputing*.
- [10] Antonio, G. P. (2015). *AN EMPIRICAL STUDY ON COLLECTIVE INTELLIGENCE ALGORITHMS*

FOR VIDEO GAMES PROBLEM-SOLVING. España:
Computing and Informatics.