



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

Centro de Ciencias Básicas
Departamento de Ciencias de la Computación

Tesis

*INTELIGENCIA ARTIFICIAL ADAPTATIVA EN VIDEOJUEGOS CON
PROCESOS ESTOCÁSTICOS*

Presenta

ERWIN BRIAN MONTES CHAPARRO

para obtener el grado de

Maestría en Ciencias, con Opción a la Computación, Matemáticas Aplicadas

Tutor

Dr. Angel Eduardo Muñoz Zavala

Comité Tutorial

Dr. Jaime Muñoz Arteaga

Mtro. Rigoberto Juárez Atic

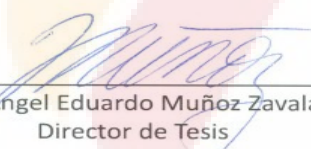
Aguascalientes, Ags., junio de 2014

M. en C. José de Jesús Ruíz Gallegos
Decano del Centro de ciencias Básicas.
Universidad Autónoma de Aguascalientes

PRESENTE

Por este conducto hago de su conocimiento que el **I.S.C. Erwin Brian Montes Chaparro**, egresado de la Maestría en Ciencias con Opción a la Computación, Matemáticas Aplicadas de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **"INTELIGENCIA ARTIFICIAL ADAPTATIVA EN VIDEOJUEGOS CON PROCESOS ESTOCÁSTICOS"**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



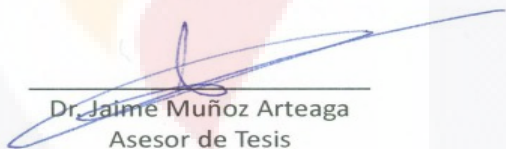
Dr. Ángel Eduardo Muñoz Zavala
Director de Tesis

M. en C. José de Jesús Ruíz Gallegos
Decano del Centro de ciencias Básicas.
Universidad Autónoma de Aguascalientes

PRESENTE

Por este conducto hago de su conocimiento que el **I.S.C. Erwin Brian Montes Chaparro**, egresado de la Maestría en Ciencias con Opción a la Computación, Matemáticas Aplicadas de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **"INTELIGENCIA ARTIFICIAL ADAPTATIVA EN VIDEOJUEGOS CON PROCESOS ESTOCÁSTICOS"**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



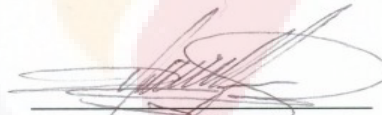
Dr. Jaime Muñoz Arteaga
Asesor de Tesis

M. en C. José de Jesús Ruíz Gallegos
Decano del Centro de ciencias Básicas.
Universidad Autónoma de Aguascalientes

PRESENTE

Por este conducto hago de su conocimiento que el **I.S.C. Erwin Brian Montes Chaparro**, egresado de la Maestría en Ciencias con Opción a la Computación, Matemáticas Aplicadas de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **"INTELIGENCIA ARTIFICIAL ADAPTATIVA EN VIDEOJUEGOS CON PROCESOS ESTOCÁSTICOS"**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



Mtro. Rigoberto Juárez Atic
Asesor de Tesis



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

Centro de Ciencias Básicas

**I.S.C. ERWIN BRIAN MONTES CHAPARRO
ALUMNO (A) DE LA MAESTRIA EN CIENCIAS
CON OPCIÓN A LA COMPUTACIÓN,
P R E S E N T E .**

Estimado (a) alumno (a) Montes:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **"INTELIGENCIA ARTIFICIAL ADAPTATIVA EN VIDEOJUEGOS CON PROCESOS ESTOCÁSTICOS"**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

A T E N T A M E N T E
Aguascalientes, Ags., 26 de mayo de 2014
"SE LUMEN PROFERRE"
EL DECANO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS



c.c.p.- Archivo.
JJRG,mjda

ÍNDICE GENERAL

ÍNDICE GENERAL.....	1
ÍNDICE DE TABLAS.....	3
ÍNDICE DE GRÁFICAS.....	5
RESÚMEN.....	7
ABSTRACT.....	9
INTRODUCCIÓN.....	11
Motivación.....	11
Contribución.....	12
Objetivo de la Investigación.....	12
Hipótesis de la Investigación.....	12
CAPÍTULO 1: ANTECEDENTES.....	13
1.1 Aprendizaje Máquina.....	13
1.2 Clasificación según la Teoría de Juegos.....	14
1.3 La Inteligencia Artificial en los Videojuegos.....	15
1.4 Inmersión.....	20
CAPÍTULO 2: CONCEPTOS BÁSICOS.....	23
2.1 Conceptos Preliminares.....	23
2.2 Procesos Estocásticos.....	24
2.3 Probabilidad Condicionada.....	24
2.4 Cadenas de Márkov.....	26
CAPÍTULO 3: PROBLEMÁTICA.....	29
CAPÍTULO 4: ESTIMACIÓN CON EL USO DE PROCESOS ESTOCÁSTICOS.....	35
4.1 Pronóstico del Estatus del Jugador.....	35
4.2 Pronóstico de los movimientos del jugador.....	36
CAPÍTULO 5: EXPERIMENTOS.....	39
5.1 Diseño de Experimentos.....	39
5.2 Condiciones de Inicialización y Pruebas.....	43
5.3 Resultados.....	44
5.4 Discusión de Resultados.....	44
CONCLUSIONES.....	47
GLOSARIO.....	49
BIBLIOGRAFÍA.....	53
ANEXOS.....	57



TESIS TESIS TESIS TESIS TESIS

ÍNDICE DE TABLAS

Capítulo 2

Tabla	Nombre	Pág.
2.1	Equivalencia de Términos	25
2.2	Matriz de Transición de n estados	27

Capítulo 3

3.1	Especificación de los tipos de ataque con sus correspondientes características	33
3.2	Especificación de los tipos de defensa y evasión con su correspondientes características.	34

Capítulo 4

4.1	Matriz de Transición para los tipos de ataque definidos	36
-----	---------------------------------------------------------	----

Capítulo 5

5.1	Matriz de transición incluyendo el estado Idle.	41
5.2	Matriz de transición desbalanceada.	41
5.3	Matriz de transición desbalanceada actualizada.	41
5.4	Estructura de combos.	42
5.5	Estructura de combos con 20 variantes.	43
5.6	Resultados de las pruebas según el grupo de Combos y el tamaño máximo de movimientos almacenados.	44

ÍNDICE DE GRÁFICAS.

Capítulo 4

Gráfico	Nombre	Pág.
4.1	Representación de la matriz de transición como un diagrama de estados.	37

Capítulo 5

5.1	Diagrama de estados final para la fase de experimentos.	40
-----	---------------------------------------------------------	----





RESÚMEN

El combate meleé es un término que aplica específicamente el combate con armas de mano, comúnmente implementado en los videojuegos en el género Hack n' Slash y en contraste a las armas de rango como el arco o las armas de fuego; en los videojuegos, el género puede incluir arqueros o especialistas de artes marciales. Los videojuegos son un área amplia en la aplicación de la Inteligencia Artificial, cuyos géneros más populares incluye los juegos de disparos (Shooters), Juegos de Rol (RPG) o de Estrategia en tiempo real (RTS). El Hack n' Slash, como parte del género de Rol brinda una experiencia única en jugabilidad debido a los parámetros que ofrecen un juego igualado entre el jugador humano y los NPCs (Non-Player characters). En la práctica, es muy poco común ver una escalada de dificultad que sea fiable, y es que incluso los juegos de última generación no los aplican en sus rutinas de Inteligencia Artificial, es decir, al comportamiento de oponentes controlados por computadora en un juego. Los procesos estocásticos son una parte de la teoría de la probabilidad que consisten en conceptos matemáticos que sirven para caracterizar una sucesión de variables aleatorias (estocásticas) que evolucionan en función de otra variable, comúnmente en el tiempo. En general, un juego puede ser representado como un proceso estocástico que evoluciona en el tiempo y, si es posible realizar una estimación de situaciones posteriores al instante actual, puede lograrse una estimación del comportamiento del contrincante en un entorno controlado. La ventaja que ofrecen los procesos estocásticos aplicados a un juego es que es posible crear una escalada automática de la dificultad para adaptar el reto que ofrece un juego hacia el jugador humano, (es decir, un juego justo) lo que da como resultado una experiencia de entretenimiento más rica frente a un juego demasiado fácil o demasiado difícil. Este trabajo propone una técnica de aprendizaje en línea que permite identificar ciertas rutinas repetitivas en el comportamiento del jugador humano y enriquecer la Inteligencia Artificial del juego. Con ayuda del teorema de Bayes y las Cadenas de Márkov se logró identificar ciertos patrones de repetición que logran satisfactoriamente estimar el movimiento siguiente que realizará el jugador. Los resultados de los experimentos indican que es posible obtener distintos intervalos de éxito en base al tamaño de la muestra capturada para el análisis. Se concluye que los procesos estocásticos en un ambiente de juego controlado, como en los juegos de combate cuerpo a cuerpo, pueden mejorar el valor de entretenimiento de un juego logrando un balance entre la complejidad de la Inteligencia Artificial del juego y la habilidad del jugador humano.



ABSTRACT

The melee combat is a term applied specifically to the weapons used in hand-to-hand combat, commonly introduced in the Hack n' Slash videogame genre and in contrast to the ranged weapons as the bow or guns; in videogames, the genre may include archers or martial artists. Videogames is a wide area for applying Artificial Intelligence, whose most popular genres are the Shooting games (shooters), Role-Playing Games (RPG) or Real-Time Strategy (RTS). Hack n' Slash, as part of the RPG genre, brings a unique experience in gameplay by the characteristics offered in a matched combat between the human player and a Non-Player Character (NPC). In practice, it's very hard to find a trustworthy Difficulty Scaling Method, and even the State-of-the-art games don't use these methods in their Artificial Intelligence routines, i.e., the behavior of the NPCs in a game. The Stochastic processes, part of the probability theory, consist in mathematical concepts to represent a sequence of random variables (stochastic) that evolve based in another variable, commonly in time. Generally, a game can be represented as a stochastic process that evolves in time and, if it is possible to estimate subsequent situations to the actual moment, the enemy behavior can be estimated in a controlled environment. The advantage of Stochastic processes applied in a game is that is possible to create an Automatic Difficulty Scaling to adapt the game challenge to the human player, (a fair game) that results in a richer game experience against a very easy or a very difficult game. This work proposes an Online Learning Technique that allows to identify certain repetitive routines in the human player behavior and improve the Game Artificial Intelligence (Game AI). Using the Bayes Theorem and Markov Chains, certain repeat patterns were identified to successfully estimate the next movement of the human player. The experiments results show that it is possible to obtain different success intervals based in the sample size for analysis. In conclusion, Stochastic processes, in a controlled game environment like the ones found in melee combat fighting games, can improve the replay value getting balance between the Game AI and the Human Player skills.



INTRODUCCIÓN.

Los videojuegos forman parte de la cultura tecnológica del siglo XXI. Hoy es más fácil encontrar a una persona que juegue algún videojuego que lo que era hace 30 años, cuando los videojuegos eran un nicho bastante pequeño comparado con hoy en día. Los videojuegos han pasado de ser un entretenimiento casual y aislado, a una de las industrias más rentables a nivel mundial. Los videojuegos han incrementado su complejidad tanto a nivel periféricos como a nivel de mecánicas de juego, invitando al jugador a tomar el papel de un personaje que puede tomar decisiones a voluntad enfrentando sus respectivas consecuencias.

El videojuego es lo más cercano a protagonizar una película sin despegarse del asiento, ser el superhéroe que salva al mundo, ser un piloto profesional de carreras, personificar a seres fantásticos de lo más recóndito de las mitologías modernas, o comandar tropas en batalla contra los más célebres estrategas en el arte de la guerra de toda la historia de la humanidad; estos son algunos ejemplos de por qué los videojuegos son una parte importante en la vida cotidiana, y son considerados como una forma más de arte debido a su manera única de contar una historia al compararse con leer un libro o ver un largometraje.

Durante las últimas dos décadas, se ha visto una evolución constante en el contenido audiovisual que involucran los videojuegos, mejorando significativamente; a pesar de las tecnologías avanzadas en gráficos y audio, los desarrolladores se empeñan poco o nada en la creación de Inteligencia Artificial (AI) en los videojuegos, también llamada *Game AI*. Desde inicios del siglo XXI, las compañías desarrolladoras y publicadoras de videojuegos han dejado de lado la calidad de la Game AI, que para el *Hardcore Gamer* constituye, hoy en día, la distinción entre un buen juego y uno mediocre. En la industria de los videojuegos, la calidad de un juego puede marcar la diferencia entre el éxito y el fracaso como estudio de desarrollo, que se ve reflejado en las ventas de un título según las recomendaciones de boca en boca y las revisiones de los medios informativos del rubro; hoy, el Hardcore Gamer busca un producto en el que pueda invertir apropiadamente su dinero y tiempo a cambio de entretenimiento de calidad que logre cubrir ese hueco en sus necesidades como jugador.

El objetivo de este trabajo de tesis es lograr un método para mejorar la calidad de la Game AI utilizando algunas técnicas de *Machine Learning* basadas en procesos estocásticos, particularmente, para crear *Non-Player Characters* (NPC) que puedan aprender de un comportamiento repetitivo del jugador que los lleve a tomar decisiones que permitan una ventaja sobre el mismo.

Motivación.

La Inteligencia Artificial aplicada en videojuegos se puede encontrar comúnmente basada en técnicas no adaptativas. Normalmente, los NPCs definen su comportamiento a base de scripts fijos que crean los desarrolladores a partir de tareas básicas que un NPC puede realizar; dichos scripts hacen que un comportamiento sea rígido e inflexible, que puede dar lugar al jugador a encontrar debilidades en los NPCs que no pueden corregir por sí mismos.

Contribución.

Una manera de ayudar a tener un NPC más complejo, es proporcionarle mecanismos que permitan tener una información casi perfecta del jugador, sin tener que caer en el Cheating [1] haciendo justa la adquisición de la información, esto es, proporcionar dicha información en un modo y cantidad similares a las que recibe el jugador humano. Dicho esto, lo que este trabajo propone es un nuevo método para contrarrestar la monotonía del jugador humano a través del uso de procesos estocásticos en base a un conjunto de posibles movimientos de combate permitidos de un juego Hack n' Slash, en un ambiente de juego simulado y controlado. Se propone un modelo adaptable en complejidad en base a los recursos de hardware disponibles, haciéndolo una herramienta escalable y robusta según las necesidades del programador.

Objetivo de la Investigación.

A. Objetivo General.

Emplear conceptos probabilísticos y métodos de procesos estocásticos, incorporando la probabilidad condicional, el Teorema de Bayes y las Cadenas de Markov para generar un método de Aprendizaje máquina no supervisado con bases en la probabilidad de transición a partir de una base de conocimientos, obtenida a través de un historial de encuentros con el jugador.

B. Objetivo Específico.

Implementar un método de evaluación que sea aplicable a un juego Hack & Slash, en donde se tratará de estimar (1) la condición del jugador en cuestión de Stamina y Health Points y (2) pronosticar los movimientos del mismo en base al movimiento actual.

Hipótesis de la Investigación.

Existe una forma de identificar patrones de comportamiento en un jugador en el ambiente controlado que proporciona el género Hack & Slash utilizando procesos estocásticos como herramienta de estimación y evaluación.

TESIS TESIS TESIS TESIS TESIS

Capítulo 1

ANTECEDENTES

Antes de profundizar en los procesos estocásticos, es necesario conocer algunos conceptos como preámbulo en materia de videojuegos.

1.1 Aprendizaje Máquina

El *aprendizaje automático* o *aprendizaje máquina* (machine learning) es una rama de la Inteligencia Artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en formas de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística, ya que las dos disciplinas se basan en el análisis de los datos. Sin embargo, el aprendizaje máquina se centra más en el estudio de la complejidad computacional de los problemas. Gran parte de la investigación realizada en aprendizaje máquina puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos.

El aprendizaje máquina tiene una amplia gama de aplicaciones, entre las cuales destacan los motores de búsqueda, diagnósticos médicos, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento de voz y lenguaje escrito, juegos y robótica [2].

1.1.1 Aprendizaje supervisado

El algoritmo produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Un ejemplo de este tipo de algoritmo es el problema de clasificación, donde el sistema de aprendizaje trata de etiquetar (clasificar) una serie de vectores utilizando una entre varias categorías (clases).

La base de conocimientos del sistema está formada por ejemplos de etiquetados anteriores. Este tipo de aprendizaje puede ser muy útil en problemas de investigación biológica, biología computacional y bioinformática.

1.1.2 Aprendizaje no supervisado

Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan solo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos. Por lo tanto, en ese caso, el sistema tiene que ser capaz de reconocer patrones para poder etiquetar las nuevas entradas.

1.1.3 Aprendizaje semisupervisado

Este tipo de algoritmos combinan los dos algoritmos anteriores para poder clasificar de manera adecuada. Se tienen en cuenta los datos marcados y los no marcados.

1.1.4 Aprendizaje por refuerzo

El algoritmo aprende observando el mundo que le rodea. Su información de entrada es la retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de prueba y error.

1.1.5 Transducción

Similar al aprendizaje supervisado, pero no construye de forma explícita una función. Trata de predecir las categorías de los futuros ejemplos basándose en los ejemplos de entrada, sus respectivas categorías y los ejemplos nuevos al sistema.

1.1.6 Aprendizaje multi-tarea

Son los métodos de aprendizaje que usan conocimiento previamente aprendido de cara a enfrentarse a problemas parecidos a los ya vistos.

1.2 Clasificación según la Teoría de Juegos.

A pesar de que la teoría de juegos se desarrolla principalmente en el ámbito filosófico, económico, biológico y social, por mencionar algunos, un juego, como al que se refiere en este trabajo, también puede entrar dentro de la categoría de juego en matemáticas aplicadas, pues consiste en un conjunto de jugadores, un conjunto de movimientos (o estrategias) disponibles para esos jugadores y una especificación de recompensas para cada combinación de estrategias [3].

A los juegos se les suele identificar según su clasificación, que incluyen juegos simétricos o asimétricos, suma cero o suma distinta de cero, simultáneos o secuenciales, con información perfecta o imperfecta, longitud finita o infinita, o siguiendo un criterio maximin y minimax [4].

En los caminos de la ciencia computacional para jugar un juego contra un jugador humano, los ejemplos abundan, siendo el más famoso el de programar una computadora para jugar ajedrez [5]. Anteriormente mencionado, se encuentra el juego de dos jugadores con suma cero, donde dos jugadores juegan alternativamente (por turnos), y el juego es visible para ambos (información perfecta) y normalmente termina con un jugador y un perdedor (incluso a veces con empate).

Los juegos de computadora pueden clasificarse en dos grupos: juegos analíticos y juegos comerciales [6]. Los juegos analíticos son juegos clásicos de tablero y cartas como el Backgammon, Damas, el

Ajedrez, el Poker, entre otros. Los videojuegos comerciales son los juegos de computadora modernos, que incluyen títulos populares como Mass Effect, The Elder Scrolls V: Skyrim, Gran Turismo, Uncharted y Warhammer 40,000: Dawn of War.

Es común encontrar que la investigación en videojuegos de computadora se ha enfocado en juegos analíticos; la meta de la investigación en un juego de computadora, es otorgarle a las computadoras una Inteligencia Artificial que logre oponentes lo más competitivos posibles para el humano. Para algunos juegos, la investigación ha logrado impresionantes resultados en juegos analíticos como en el Ajedrez [7], Damas [8] y Otello [9], derrotando a campeones del mundo.

A partir del inicio del siglo XXI, la investigación referente a los juegos de computadora se ha extendido para abarcar los juegos comerciales [10]. Una inspección más cercana muestra que los juegos analíticos difieren de los juegos comerciales en muchas características [6], de las cuales destacan: Clasificación por teoría de juegos, origen de la complejidad, requerimiento computacional, ritmo, drama, inversión del rol, arte, habilidades del jugador y meta en el juego.

Debido al contraste en las características de los juegos analíticos y los juegos comerciales, la mayoría de los trabajos de investigación enfocados a los juegos analíticos no son aplicables a los juegos comerciales. Hay muchos problemas en la Game AI de los juegos comerciales que la investigación de juegos analíticos no abarca, como el Pathfinding [11], Inteligencia Artificial Adaptable [1], Adaptación basada en preferencias [12], por mencionar solo algunos ejemplos.

El término "Juego Comercial" podría ser engañoso, debido a que los juegos analíticos también pueden ser explotados comercialmente. Un término que se puede encontrar en la literatura es el de "Juego de computadora Interactivo"[6], pero debido a que todos los juegos de computadora son interactivos, hay una desorientación mucho mayor. Para este trabajo de tesis, se hará referencia a los juegos comerciales con el término "videojuego" para cuestiones prácticas .

1.3 La Inteligencia Artificial en los Videojuegos.

Los videojuegos del siglo XXI siguen utilizando las mismas técnicas de comportamiento de hace más de 20 años, que tienen como métodos fundamentales los autómatas finitos con ciertas derivaciones como las máquinas de estado jerárquicas [13] y pocos han sido los desarrolladores que se han atrevido a abandonar los autómatas para entrar a otro nivel de sistemas de control y toma de decisiones como los árboles de comportamiento [14] que combinan las mejores funcionalidades de scripts lineales y máquinas de estado reactivas.

A pesar de que existen títulos que pueden tener una Game AI compleja, puede llegar a ser malinterpretada o poco creíble [15], incluso en ocasiones la complejidad se podría tomar como si la Game AI estuviera utilizando *cheats* para conocer más de la situación del jugador como lo hizo en sus inicios la saga de Half-Life [1] o como lo utiliza Left 4 Dead para dar prioridad a los jugadores humanos sobre los NPCs amigos [16]. Es un gran problema el intentar mostrar una Game AI creíble sin llegar a parecer inhumana, o rayando en la perfección; si bien un equipo de fuerzas especiales tiene un

mínimo margen de error al entrar en acción en una situación de riesgo de la vida real, en los videojuegos puede llegar a frustrar al jugador que busca un reto que al final sea posible vencer [17].

Existen herramientas en los motores de juego que facilitan la concentración de esfuerzo en el apartado de Game AI, pues hay herramientas que pueden encargarse tanto de las animaciones del personaje [15], como de su desplazamiento dentro de un escenario [16]; todo esto hace que la parte acciones básicas de un NPC exista de manera casi automática, dejando la parte de evaluación y toma de decisiones a criterio del programador.

1.3.1 Inteligencia Artificial por género.

Los videojuegos como campo de aplicación de la Inteligencia Artificial buscan simular el comportamiento de entes vivientes (según el género y el tipo de personajes). Existe variedad de estudios que proponen mejoras a varios tipos de juegos pero, en su mayoría, no han sido utilizados para juegos a nivel comercial [12], lo que hace que haya una gran área de oportunidad en el rubro de la Game AI.

La Inteligencia Artificial, como definición académica o a veces referida como “IA Tradicional” [18][19], y es aquella que utiliza los conceptos que se encuentran en la naturaleza y son adaptados a algoritmos computacionales para la solución de problemas, tales como la optimización mediante Colonia de Hormigas, las Redes Neuronales, Enjambre de Partículas, Algoritmos Culturales y Técnicas heurísticas híbridas, por mencionar algunos ejemplos. La principal característica de la IA Tradicional, es el hecho de que se utilizan todos los recursos posibles en un sistema de cómputo para realizar sus operaciones.

Por otro lado, la Inteligencia Artificial en los videojuegos tiene como meta el lograr el diseño de agentes que provean la ilusión de inteligencia utilizando recursos limitados en comparación con la IA Tradicional; la experiencia se enriquece, mostrando a la percepción del jugador, pistas audibles o visuales donde se pueda interpretar fácilmente lo que los agentes están haciendo [18]; debido a ésto, se requieren técnicas implementadas a través de rutinas que tengan un balance entre complejidad y rendimiento con una cantidad de recursos proporcionalmente baja en comparación con el resto del juego como un todo.

A continuación se citan algunos de los géneros más conocidos:

Shooters.

El género que ha tomado más fuerza en los últimos años han sido los videojuegos de tipo Shooter como Half-Life [1] o la saga de Halo [13], en la que se requiere la evaluación y la toma de decisiones en línea sobre el jugador para dar la impresión de que, el mismo jugador, se está enfrentando a criaturas inteligentes y no a NPCs que cuentan con un scripting estático y no adaptable. En este género es en el que más se recurre a la Estupidez Artificial [17], definida por errores cometidos de manera intencional por los NPCs para semejar al ser humano y su imperfección [13]. De este género se derivan: First Person Shooter (FPS), que emulan la vista a través de los ojos del personaje; los Third Person Shooter, que tienen un ángulo de cámara a espaldas del protagonista. De ambos sub-géneros existen variaciones en el gameplay dependiendo de cada videojuego, pero es fácil identificarlos a simple vista.

Real-Time Strategy.

Uno de los géneros más exclusivos de videojuegos para PC, por la cantidad de instrucciones disponibles, es el Real-Time Strategy (RTS) en el que los jugadores comandan una cantidad de unidades bastante superior al de otros géneros para seguir la voz de mando del jugador humano. Buena parte de la evaluación del juego se lleva a cabo a través de análisis estadístico a partir de la distribución y cambio de ciertos parámetros específicos que se utilizan en una sola función de evaluación [1][12]. En este género es relativamente sencillo obtener una adaptación óptima al modelo del jugador, debido a que las partidas en este tipo de juegos duran al menos 30 minutos, extendiéndose a partidas mayores de 60 minutos o más, según sea el caso. Debido al ritmo de juego, es posible acceder a estrategias anteriormente almacenadas según una base de conocimientos previamente establecida. Este género no cuenta con divisiones marcadas, pues las diferencias entre títulos se basan más en la cantidad de unidades bajo el mando del jugador o la cantidad y variedad de habilidades de las unidades según su tipo.

Rol-Playing Game.

El género de Rol-Playing Game (RPG), es una clasificación de juegos bastante particular, ya que se utiliza un personaje principal, como pasa en los shooters, pero con una cantidad de movimientos y habilidades superior que provoca una elevación en la complejidad según el avance en el juego. La mayoría de los NPCs en este género están basados en scripts fijos, en los cuales el jugador fácilmente puede encontrar exploits o incongruencias, esto debido a que los programadores no tienen la capacidad de considerar todas las situaciones y tomas de decisiones posibles en una escena en específico [1]. En este género existen varios subgéneros, dentro de los cuales destacan (1) los Massive Multiplayer Online RPG (MMORPG), y en algunas ocasiones algunos juegos del tipo (2) Hack n' Slash, dentro de los cuales no se ha visto un comportamiento adaptable de NPCs según las acciones del jugador.

1.3.2 La Inteligencia Artificial en la Industria.

Los videojuegos del género Role-Playing Games (RPG) son comúnmente conocidos por el incremento en las habilidades del personaje que maneja el jugador humano. Anteriormente los videojuegos RPG incorporaban, en su mayoría, mecánicas de juego por turnos como han sido desde hace mucho tiempo los juegos de Final Fantasy, en el que cada oponente controla a varios personajes, capaces de realizar sólo un ataque por turno y sin límite de tiempo que permitía al jugador humano pensar su siguiente movimiento como si fuera un tablero de ajedrez. La principal característica de éste género fue que los personajes obtenían puntos de experiencia tras cada combate (Experience Points o XP) y lograban desarrollar habilidades más poderosas o perfeccionar las ya aprendidas.

Por otro lado, el género Hack'n Slash (H&S) fue siempre un género que incorporaba una mecánica basada en mayor parte en el combate cuerpo a cuerpo y con armas de filo como las espadas y las hachas, combinadas con escudos y otros artefactos contemporáneos como el arco y la flecha o la ballesta y la saeta [20]. Con el paso del tiempo los juegos de este género evolucionaron trayendo experiencias nuevas en franquicias conocidas como las sagas de *Knights of the Old Republic* o *Jedi Academy* desarrolladas por Lucas Arts y basadas en el universo de Star Wars creado por George Lucas,

donde se cambiaban los arcos y las espadas de metal por pistolas de rayos y sables de luz, incluyendo las habilidades que podían describir a un Caballero Jedi o un Lord Sith.

La combinación de H&S y el RPG convencional se integraron en lo que después se conoció como Action RPG (ARPG) que incorporaba el sistema de combate dinámico del H&S con la obtención de XP de los RPGs. Actualmente los juegos H&S han cambiado de orientación volviéndose juegos de tipo *Beat'em up* [21] donde la sensación de un combate cuerpo a cuerpo de manera balanceada ha dejado de existir.

Pocos son los juegos que continúan con lo que alguna vez fue el género H&S y en su mayoría han cambiado a juegos de toque fantástico con enemigos marcadamente inferiores a excepción de los enemigos especiales o jefes de nivel. Juegos como God of War, Darksiders, Devil May Cry y Dante's Inferno, que son considerados H&S, colocan en una ventaja abrumadora al jugador con habilidades que pueden ser utilizadas en conjunto a manera de *combo* sin darle oportunidad a los contrincantes de defenderse, y así logrando que la rejugabilidad basada en la dificultad del juego sea por los parámetros que definen a los NPCs y no el cambio de táctica del jugador ante una situación distinta, porque los NPCs no son capaces de prevenir las acciones repetitivas del jugador.

Tanto los juegos ARPG como los actuales, denominados, Hack'n Slash sólo funcionan a través de validaciones condicionales que sólo toman en cuenta parámetros pequeños y desperdician otra información valiosa que puede hacer que el juego pueda balancearse entre el jugador y los NPCs.

Los RPGs el día de hoy son capaces de cambiar puntos claves de la trama basadas en decisiones que toma el jugador a través de las conversaciones que toma con los NPC que conviven con él, como es el caso de la saga de Mass Effect, donde el jugador puede decidir su destino y el de otros personajes a partir de su actitud bondadosa o vengativa durante todo el juego afectando hechos posteriores.

Normalmente, a nivel comercial, la mayoría de los métodos de Inteligencia Artificial más sobresalientes, son aplicados a juegos de tipo Shooter (FPS y TPS) como la saga de Halo (Bungie Studios) [13], Crysis (Crytek) [15] o Grand Theft Auto (Rockstar Games) [22], pero que a pesar de estar entre los más sobresalientes en la industria, no muestran algún método que sea eficiente para descifrar la conducta del jugador. Los Shooters son los juegos en donde más se desperdicia información que puede ser valiosa para contrarrestar al jugador en ciertas situaciones. Para mostrar parte de la información de las acciones que se llevarán a cabo por parte de los NPCs, éstos suelen avisar intencionalmente con frases que hacen que el jugador tenga una idea de qué está ocurriendo, como "Recargando el arma", "Estoy Herido", "¡Granada!", entre muchas otras.

1.3.3 La Inteligencia Artificial en la Investigación.

Existe una gran cantidad de estudios para la Inteligencia Artificial de los juegos que incluyen todo tipo de géneros, pero, debido a las diferencias de Gameplay, es difícil que todas las técnicas se puedan aplicar en todos ellos, pues manejan situaciones con parámetros distintos en número y características; sin importar estas situaciones, existen algunas técnicas que pueden ser aplicadas a varios géneros, como RTS y RPGs, debido al control del entorno con el que cuenta el programador a diferencia de

situaciones más abiertas y menos controlables como lo es el ambiente de un FPS.

A continuación se mencionan algunos artículos relacionados con la adaptación de la Game AI al jugador humano utilizando distintas técnicas e incluyendo sus principales características y los resultados que se obtuvieron.

Uno de los principales autores en el campo de la Inteligencia Artificial Adaptativa es Pieter Spronk [6] quien ha publicado y ayudado en varias investigaciones sobresalientes en el tema.

El estudio de *“Extraversion in Games”* [23] demuestra que existe gran cantidad de información que puede ser obtenida tan sólo tomando en cuenta las decisiones del jugador mientras se desarrolla el juego. De ésta manera se obtiene un perfil del jugador que permite incrementar el entretenimiento en base al comportamiento altruista o egoísta del mismo u obteniendo el modelo del jugador para ajustar la línea argumental a sus preferencias. Aún obteniendo el perfil del jugador, se sigue desaprovechando mucha información que puede ser utilizada para llevar a cabo acciones pertinentes que permitan ajustar el juego según las acciones que lleva a cabo el jugador mientras intenta encontrar la manera más sencilla de derrotar a los NPCs buscando exploits.

En *“Rapid Adaptation of Video Game AI”* [1] se plantea una adaptación utilizando el razonamiento basado en casos en el motor para juegos RTS llamado Spring. Se recolectan las observaciones hechas por el enemigo hacia el jugador y se extrae una base de casos. Se contienen todas las observaciones relevantes sin redundancias, con registro cronológico y estructuradas en un formato standard para un acceso rápido. Por la naturaleza del género en particular, se toman en cuenta variables dentro del juego como la cantidad de unidades y edificios de cada tipo y nivel con las que cuenta el jugador según las observaciones realizadas en el transcurso de tiempo de la partida para el análisis “online”. El mecanismo de adaptación consiste en indexar los juegos recolectados haciendo clustering de manera “offline” agrupando observaciones similares. Se establece la estrategia más parecida a la que realiza el jugador humano y se inicializa la IA del juego con una estrategia observada contra el oponente en donde los parámetros sean más parecidos en la base de casos. Se seleccionan los juegos que satisfacen más el criterio meta, y después se guardan las observaciones relevantes. Los experimentos se llevan a cabo en contra de “Inteligencias Artificiales” que contiene el motor por defecto para su comparación. Los resultados demuestran que el 80% de las pruebas realizadas existe la victoria para el método propuesto y que logra adaptarse relativamente rápidamente durante una partida.

Otra implementación aplicada a juegos del género RTS, pero más relacionada con la actitud de un NPC ante el jugador, es el trabajo de *“Player Modeling in Civilization IV”*[12], en donde se busca encontrar el Modelo de jugador en base a las estrategias aplicadas por el NPC que representa el comandante y las órdenes de su respectivo ejército. En comparación con el trabajo de adaptación del trabajo sobre el motor Spring [1], en éste caso en lugar de nivelar el juego, basándose en la cantidad de unidades y edificios, se desea encontrar un modelo que defina la postura del jugador basándose en sus preferencias y actitudes ante el comandante enemigo.

En *“Difficulty Scaling of Game AI”* [24] se proponen nuevos tipos de escalación de la dificultad en el juego del Género ARPG llamado Baldur’s Gate. La investigación se lleva a cabo con Scripting Dinámico

[25] utilizando aprendizaje “online” para reparar automáticamente debilidades encontradas en los scripts que son explotadas por el jugador o para adaptarse mejor a las tácticas del jugador. Se centra principalmente en técnicas de refuerzo de aprendizaje adaptadas para utilizarse en videojuegos debido a que las técnicas regulares de refuerzo de aprendizaje no son eficientes en el aprendizaje online en juegos. Se definen macros compuestas por varios movimientos encadenados. Se evalúa cada una de las Macros en tiempo de ejecución para determinar si fue exitosa tomando en cuenta si el NPC ganó o perdió, si murió o sobrevivió, la cantidad de vida restante y el monto total de daño hecho al enemigo.

Cada Macro tiene un peso asignado por defecto y éste se va incrementando o decrementando dependiendo del peso que se va asignando después de una ejecución y también dependiendo del método.

Las técnicas utilizadas para escalar la dificultad son:

- High Fitness Penalising: Se dan mayores recompensas a resultados mediocres y se penalizan a los mejores resultados.
- Weight Clipping: Se ajusta un peso máximo para las macros, haciendo que otras puedan ser candidatas a ser elegidas para su ejecución.
- Top Culling: Las macros con mayor peso se eliminan cuando pasan del límite máximo de peso, dando oportunidad a otras macros de tener oportunidad de ocurrir. Una vez que pasa cierto lapso, la macro que ha sido capada puede volver a utilizarse.

Para llevar a cabo las pruebas se compararon los 3 métodos propuestos contra la IA que incorpora el juego por defecto. Los resultados muestran que el mejor método es el “Top Culling”, aunque podrían utilizarse en conjunto en un mismo juego para definir distintos tipos de aprendizaje o incluso niveles de dificultad distintos.

Como anteriormente ya se había mencionado, muy pocas técnicas del área de investigación en el rubro son aplicadas a juegos comerciales.

Este trabajo propone una herramienta que no sólo permita encontrar un balance justo en el juego entre los NPCs y el jugador, sino que también tenga la posibilidad de aprender las técnicas a las que más recurre el jugador de manera online, y calcular la probabilidad de que el jugador se encuentre en circunstancias de riesgo, ya sea por su nivel de estamina o por su nivel de salud. Tanto el nivel de estamina como de HP pueden ser calculados de manera aproximada para que exista equidad de circunstancias, pues es difícil para el jugador saber en qué niveles se encuentra el NPC a excepción de que exista algún indicador explícito en la pantalla durante el juego.

1.4 Inmersión

La inmersión es definida como medida general de entretenimiento, e implica la sensación de adentrarse en un videojuego haciendo que el jugador mantenga el interés en el mismo [1]. Este concepto ha evolucionado a la par de los videojuegos desde sus inicios gracias a las tecnologías actuales tanto en (1) gráficos, (2) sensibilidad e interacción y (3) sonido. El realismo en el apartado

TESIS TESIS TESIS TESIS TESIS

gráfico logra la sensación cercana a protagonizar una película mientras el jugador define el destino de los personajes. La parte de sensibilidad e interacción va relacionado con cuestiones sensoriales, que es donde entran los modos de vibración de controles o mandos (Vibration Feedback y Force Feedback) hasta los controles que involucran el movimiento natural como el *Wii-mote* del *Nintendo Wii*, *Microsoft Kinect* de *Xbox 360* y *Sony Playstation Move* del *Playstation 3*, que han logrado que el jugador deje su asiento para participar más naturalmente en la acción del juego. Por el lado del sonido se encuentran tecnologías de audio ambiental como EAX (Environmental Audio Extension) que permite la reproducción de hasta 128 voces a la vez en su versión 5.0 (referencia oficial EAX), haciendo que la percepción del entorno logre una experiencia más inmersiva incluso al nivel de materiales (el sonido pasa con diferente intensidad si se encuentra tras una pared de concreto o una pared de madera); todo esto se une a la capacidad de los equipos de audio más sofisticados con hasta 8 canales de audio (7.1 canales) para llevar la experiencia de audio y la inmersión a un nivel superior.

Un *gameplay* elaborado también es pieza clave en el concepto de inmersión, pues si no se logra una concentración en el videojuego, el jugador puede perder rápidamente interés en él, haciendo que la experiencia no sea del todo grata o entretenida.

Algunos de los ejemplos de videojuegos recientes con un alto nivel de inmersión son: *Amnesia: The Dark Descent* (Frictional Games – 2009), en el que los gráficos pasan a un segundo plano para dar lugar a la oscura atmósfera creada entre las sobras de un antiguo castillo en una inigualable integración con el audio; *Dear Esther* (*The Chinese Room* – 2012), en el que la sensación de soledad en unión con los hermosos paisajes y la narrativa única logran que sea un juego como pocos se han visto en la última década.



Capítulo 2

CONCEPTOS BÁSICOS.

Los videojuegos pueden ser interpretados tomando las acciones del jugador humano hacia el NPC como sucesos con cierto nivel de aleatoriedad, lo que permite un análisis más detallado utilizando términos estadísticos que incluyen la Probabilidad y los procesos estocásticos.

2.1 Conceptos Preliminares

La *Probabilidad* es la medida cuantitativa promedio de la cual se obtiene la frecuencia de un suceso determinado mediante la realización de un experimento aleatorio, del que se conocen todos los resultados posibles, bajo condiciones suficientemente estables [26].

Un *espacio muestral* consiste en un conjunto de todos los posibles resultados individuales de un experimento aleatorio. Un *suceso* o *evento* es cualquier subconjunto del espacio muestral, llamándose a los sucesos que contengan un único elemento *sucesos elementales*.

La *Teoría de la Probabilidad* se usa extensamente para sacar conclusiones sobre la probabilidad discreta de sucesos potenciales y la mecánica subyacente discreta entre sistemas complejos.

La *Frecuencia* es la cantidad de veces que se repite un determinado valor de la variable, que puede ser representada en Histogramas y Diagramas de Pareto. La *Frecuencia Absoluta* es el promedio de una suma predeterminada; consiste en saber cuál es el número o símbolo de mayor equivalencia estadística X_i , en otras palabras, el número de veces que el valor aparece en el estudio (moda). La *Frecuencia Relativa* es la división de la frecuencia absoluta con respecto al tamaño de la muestra.

Una *Sucesión* es una lista ordenada de objetos, cada uno de ellos denominado término (elemento o miembro) de la sucesión, y al número de elementos ordenados (posiblemente infinitos) se le denomina longitud de la sucesión. Formalmente es una función sobre el conjunto de los números naturales. Si una sucesión tiene límite, se dice que es una sucesión convergente, y que la sucesión converge o tiende al límite; en el caso contrario, la sucesión es divergente.

Una *variable estadística* o *variable* es una característica que al ser medida en diferentes individuos es susceptible de adoptar diferentes valores. Una *variable discreta* es aquella que representa separaciones o interrupciones en la escala de valores que puede tomar; estas separaciones o interrupciones indican la ausencia de valores entre los distintos valores específicos que la variable puede asumir. Una *variable continua* puede adquirir cualquier valor dentro de un intervalo especificado de valores (como la masa en kg o la longitud en metros); solamente se está limitado por la precisión de la medición, en teoría permiten que exista un valor entre dos variables.

En Probabilidad, una *variable aleatoria* o *variable estocástica* es una variable estadística cuyos valores

se obtienen de mediciones en algún tipo de experimento aleatorio. Formalmente, una variable aleatoria es una función que asigna eventos. Los valores posibles de una variable aleatoria pueden representar los posibles resultados de un experimento aún no realizado, o los posibles valores de una cantidad cuyo valor actualmente existente es incierto. Intuitivamente una variable aleatoria puede tomarse como una cantidad cuyo valor no es fijo pero puede tomar diferentes valores; se usa una distribución de probabilidad para describir la probabilidad de que se den los diferentes valores.

Una *Sucesión de Variables Aleatorias* es una ley o criterio que a cada número natural le asocia una variable aleatoria. Denotado por X_n , a la variable aleatoria que la ley en cuestión asocia a un número natural n para referirnos a una sucesión, se escribe $\{X_n\}=\{X_1, X_2, X_3, \dots, X_n\}$ con respecto al tiempo, con el tiempo como una medida de cambio.

2.2 Procesos Estocásticos

Un *Proceso Estocástico* es una sucesión de variables aleatorias cuyas características pueden variar a lo largo del tiempo. Cada variable aleatoria del proceso tiene su propia función de distribución de probabilidad y, entre ellas, pueden estar correlacionadas o no.

Algunos ejemplos de procesos estocásticos incluyen: las señales de telecomunicación; los índices de la bolsa de valores segundo a segundo; el tiempo de espera en una fila; el clima, que conforma un gigantesco cúmulo de procesos estocásticos interrelacionados (velocidad del viento, humedad en el aire, presión atmosférica, entre otros) que evolucionan en el espacio y tiempo.

Un proceso estocástico se puede definir equivalentemente de dos formas diferentes:

- Como un conjunto de realizaciones temporales y un índice aleatorio que selecciona una de ellas.
- Como un conjunto de variables aleatorias X_t indexadas por un índice t dado que $t \in T$ con $T \subseteq \mathbb{R}$

T puede ser continuo si es un intervalo (el número de sus valores es limitado) o discreto si es numerable (solamente puede asumir determinados valores). Las variables aleatorias X_t toman valores en un conjunto que se denomina *espacio probabilístico*.

2.3 Probabilidad Condicionada

En teoría de conjuntos, se denomina *suceso* a todo subconjunto del espacio muestral $A \subseteq \Omega$. Las propiedades de una función de probabilidad en teoría de conjuntos son:

$$\begin{aligned} P(A) &> 0 \\ P(\Omega) &= 1 \\ P(\bar{A}) &= 1 - P(A) \\ P(\emptyset) &= 0 \end{aligned}$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$

Si $A \subseteq B$ entonces $P(A) \leq P(B)$

Existe equivalencia entre los términos del cálculo de probabilidades y la teoría de conjuntos, como se muestran en la *Tabla 1.1*.

Cálculo de Probabilidades	Teoría de Conjuntos
Suceso Seguro (Espacio Muestral)	Conjunto Universal
Suceso Elemental	Punto del Conjunto Universal
Suceso	Subconjunto
Sucesos Incompatibles	Conjuntos Disjuntos
Unión de Sucesos	Unión de Conjuntos
Suceso Imposible	Conjunto Vacío
Suceso Contrario	Conjunto Complementario
Intersección de Sucesos	Intersección de Conjuntos
Sistema Completo	Partición

Tabla 2.1 Equivalencia de términos.

Es la probabilidad de que ocurra un evento A sabiendo que también sucede otro evento B . Si $P(B) > 0$, la probabilidad condicionada de que se realiza B , $P(A/B)$ viene definida por el cociente:

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

Analógicamente se define la probabilidad condicionada de B respecto a A . Utilizando conjuntamente ambos resultados se obtiene que:

$$P(A \cap B) = P(A/B)P(B) = P(B/A)P(A)$$

El *Teorema de la probabilidad total* es considerado un experimento que se realiza en dos etapas:

1. Se suponen los posibles sucesos A_1, A_2, \dots, A_n que constituyen un sistema completo, de tal forma que son conocidas las probabilidades *a priori* $P(A_1), P(A_2), \dots, P(A_n)$.
2. Los resultados posibles B_j tienen probabilidades desconocidas que dependen de lo que ocurra en la primera etapa. Si se conocen las probabilidades condicionadas $P(B/A_i)$ para cada cierto suceso B y cada A_i se verifica que:

$$P(B) = \sum_{i=1}^n P(B/A_i)P(A_i)$$

El *Teorema de Bayes* establece que se puede plantear la probabilidad de que se haya dado un determinado suceso sabiendo que como resultado final del experimento se ha obtenido otro determinado suceso.

En la misma hipótesis del teorema anterior, se tiene que:

$$P(A_k/B) = \frac{P(B/A_k)P(A_k)}{\sum_{i=1}^n P(B/A_i)P(A_i)}, k=1, \dots, n$$

Se expresan las probabilidades *a posteriori* $P(A_i/B)$ en función de las verosimilitudes $P(B/A_i)$.

2.4 Cadenas de Márkov

La *Propiedad Markoviana* se refiere a la propiedad de ciertos procesos estocásticos por la cual “carecen de memoria”, lo que significa que la distribución de probabilidad del valor futuro de una variable depende de su valor presente, pero es independiente de la historia de dicha variable.

Sea $\{X_n: n \geq 0\}$ un proceso estocástico discreto, i.e., cada X_n es una variable aleatoria discreta que forma una secuencia de variables aleatorias. Si la distribución de probabilidad condicional de X_{n+1} en estados pasados es una función de X_n por sí sola, entonces:

$$P(X_{n+1}=j / X_0=i_0, X_1=i_1, \dots, X_n=i_n) = P(X_{n+1} / X_n=i_n) = p_{i,j}(n)$$

Donde X_i es el estado del proceso en el instante i . La identidad mostrada es la propiedad de Márkov.

La *Probabilidad de Transición* ocurre cuando $p_{i,j}(n)$ es la probabilidad de que el proceso, estando en el estado i y en el tiempo n , pase al estado j en el instante siguiente.

Cuando $p_{i,j}(n) = p_{i,j}$ (esto es, no depende de n) se dice que las probabilidades de transición son estacionarias.

Las probabilidades de transición definen la *matriz de transición* $P = [p_{i,j}]$ que satisface:

1. $p_{i,j} \geq 0$ para todo i, j
2. $\sum_j p_{ij} = 1$ para todo i

A continuación un ejemplo de una matriz de transición con n estados:

		Tiempo $n+1$			
		Estado 0	Estado 1	...	Estado n
Tiempo n	Estado 0	$p_{0,0}$	$p_{0,1}(n)$...	$p_{0,n}(n)$
	Estado 1	$p_{1,0}(n)$	$p_{1,1}$...	$p_{1,n}(n)$
	\vdots	\vdots	\vdots	\vdots	\vdots
	Estado n	$p_{n,0}(n)$	$p_{n,1}(n)$...	$p_{n,n}$

Tabla 2.2 Matriz de Transición de n estados.

Una *Cadena de Márkov* es un tipo especial de proceso estocástico discreto en el que la probabilidad de que ocurra un evento depende del evento inmediatamente anterior. La dependencia del evento anterior distingue a las cadenas de Márkov de las series de eventos independientes como tirar una moneda al aire o un dado. Matemáticamente se define como un proceso estadístico discreto que cumple con la propiedad Markoviana, es decir, si se conoce la historia del sistema hasta su instante actual, su estado presente resume toda la información relevante para describir en probabilidad su estado futuro.

Se dice que una Cadena de Márkov es *homogénea* si la probabilidad de ir del estado i al estado j en un paso no depende del tiempo en el que se encuentra la cadena, esto es:

$$P(X_n = j | X_{n-1} = i) = P(X_1 = j | X_0 = i) \text{ para todo } n \text{ y para cualquier } i, j$$

Si para alguna pareja de estados y para algún tiempo n la propiedad antes mencionada no se cumple, diremos que la Cadena de Márkov es *no homogénea*.

La probabilidad de ir de un estado i al estado j en n unidades de tiempo es

$$p_{ij}^{(n)} = P(X_n = j | X_0 = i)$$

En la probabilidad de transición de un paso se omite el superíndice, de modo que queda

$$p_{ij} = P(X_1 = j | X_0 = i)$$

Las probabilidades de transición en n pasos satisfacen la ecuación de Chapman-Kolmogorov, esto es, para cualquier k tal que $0 < k < n$ se cumple que

$$p_{ij}^{(n)} = \sum_{r \in E} p_{ir}^{(k)} p_{rj}^{(n-k)}$$

donde E denota el espacio de estados.



Capítulo 3

PROBLEMÁTICA.

Un juego es un contexto con reglas entre adversarios que intentan conseguir objetivos; es una actividad entre dos o más personas con capacidad de tomar decisiones que buscan alcanzar objetivos dentro de un contexto limitado [27].

Un videojuego es una prueba mental, llevada a cabo frente a una computadora de acuerdo con ciertas reglas, cuyo fin es la diversión o esparcimiento, mientras que un juego serio se define como una prueba mental de acuerdo a reglas específicas que usa la diversión como modo de formación gubernamental o corporativo, con objetivos en el ámbito de la educación, sanidad, política y comunicación estratégica [28].

Comúnmente, un videojuego no entra en la clasificación de los juegos con información perfecta en materia de teoría de juegos, pues el NPC tiene la capacidad de saber el estado y posición de un jugador en toda situación, mientras que el jugador no cuenta con esa misma información en la mayoría de los casos y sin recurrir a Cheats.

Uno de los ejemplos más marcados son los juegos de peleas, donde los NPCs siempre saben los movimientos del jugador en cuanto éste presiona los botones en su mando (o control), haciendo que el nivel de dificultad sea considerablemente alto y de cierta manera injusta debido a la cantidad de información con la que cuenta el jugador; por otro lado, debido a que los NPCs están diseñados en base a Autómatas finitos, el jugador puede encontrar rápidamente las flaquezas en el modo de juego del NPC, pues el NPC solamente responde a acciones según condiciones que ocurren durante un combate y que fueron definidas de manera “fija” por el programador, tornando el juego en una experiencia monótona debido a la capacidad del jugador humano de encontrar una solución al problema rápidamente.

Otro ejemplo está en el género de RTS, donde el NPC en el rol del comandante siempre está consciente de la posición, cantidad y estado de todas las unidades del jugador humano [19], incluso cuando los programadores implementan el concepto de *Fog of War*, para aparentar que no se sabe el estado exacto de las unidades del bando contrario; ésto puede apreciarse fácilmente en una partida de Starcraft (Blizzard – 1998) y aún en títulos más recientes como World in Conflict (Massive Entertainment – 2007), haciendo de la experiencia algo nefasto por la injusta diferencia en la cantidad de información que se tiene entre ambos comandantes durante el juego.

Muchos juegos son capaces de mostrar un comportamiento complejo de NPCs, que si se analizaran utilizando el Test de Turing [29], podrían pasarlo en algunos casos aislados, pero la cantidad de recursos disponible para el programador y las técnicas de Inteligencia Artificial se han visto limitados por el hardware de las videoconsolas y sus recortadas características de hardware destinadas principalmente al cálculo de propósito general y los gráficos [30].

Existen casos en los que el desarrollador aprovecha toda la información que genera el motor para utilizarlo como entrada en la función de evaluación que determinará la siguiente acción a realizar contra el jugador como es el caso de la saga de Half-Life [1] y que permite a los NPCs la toma de decisiones que aparente un comportamiento inteligente pero sin abusar el cheating a través de esta fuente de información.

Como anteriormente se ha mencionado, este trabajo de investigación se centra en el género Hack & Slash, que tiene como particularidad el combate meleé que consiste en un combate entre 2 individuos, normalmente en igualdad de condiciones. Un ejemplo básico de un combate meleé en igualdad de condiciones puede ser un combate de Box, en el que ambos participantes buscan la manera de conectar más golpes a su contrincante para ganar un combate; si vemos este ejemplo aplicado en su totalidad, cada combatiente busca leer los movimientos del contrario para saber en qué momento atacar, defenderse o esquivar un ataque, dando ventaja táctica que permitirá conectar otros ataques o combinaciones de ataques.

Del Combate Igualado al Hack n' Slash.

Si vemos en retrospectiva algunos de los juegos que iniciaron con el concepto de un combate igualado, podemos ver, como por ejemplo Boxing (Activision – 1980) [31] en el que dos adversarios (2 jugadores humanos o un jugador humano contra la máquina) simulaban un combate de Box visto desde arriba con un sistema de combate sencillo y calificado por puntos, en el que se sumaban puntos al jugador dependiendo de la cercanía del golpe en un combate de 2 minutos; si bien eran mecánicas bastante simples, la igualdad de condiciones estaba presente, aunque el NPC no tuviera capacidad de aprender. Los controles permitían a los jugadores desplazarse en los ejes {X,Y} de un cuadrilátero con solamente un botón de acción que representaba el golpe y que seleccionaba automáticamente uno de los brazos dependiendo de la altura en Y con respecto a la altura en la que estaba el rostro del contrincante.

Un ejemplo posterior al combate igualado, puede apreciarse en el título Karateka (Jordan Mechner – 1984) para Apple II y la consola Nintendo Entertainment System (NES) [32], con gran realismo en sus animaciones para la época (juego precursor del primer Prince of Persia) y con un sistema de combate básico, pero igualado, uniendo elementos que pueden encontrarse en juegos Beat'em Up y muchos juegos de peleas modernos. Se tenían dos tipos de ataque que incluían golpes y patadas, con variaciones en la altura de cada movimiento (al rostro, al torso, a las piernas), dándole variabilidad y cierto nivel de aleatoriedad que incrementaba la dificultad del juego, además de no contar con opción de guardar avances, haciendo que fuera necesario acabar el juego en una sola sesión. En la época en la que este juego se creó, comenzaban a apreciarse elementos de complejidad en los personajes y máquinas de estados básicas para simular un comportamiento que contrarrestara al jugador de alguna manera, pero sin contar con aprendizaje y análisis del mismo. Los controles del juego incluían desplazamiento unidimensional (pues solo se movía hacia atrás o adelante) y dos distintos botones que podían accionar patada o golpe, cuya altura podía ser modificada utilizando los botones de dirección en el control.

Tiempo después comenzó la época dorada de los Beat'em up, que tenían la particularidad de

enfrental a jugador a un número grande de enemigos que tenían una capacidad de combate y energía inferior que agregaban dificultad según la cantidad de enemigos o la etapa del juego en la que se encontraban. El género Beat'em up llegó a la popularidad con juegos como Double Dragon (Taito – 1987), Teenage Mutant Ninja Turtles (Konami – 1989), Golden Axe (Sega – 1989), Final Fight (Capcom – 1989), Captain America and the Avengers (Data East – 1991), Battletoads (Rare – 1991) y2 posteriormente, en sagas más recientes como Devil May Cry (Capcom – 2001), Ninja Gaiden (Tecmo – 2004 como spin-off del juego original de NES), God of War (Sony – 2005), Castle Crashers (The Behemoth – 2008), Bayonetta (Platinum Games – 2009), Metal Gear Rising (Platinum Games – 2013) por mencionar los más representativos. Varios de estos títulos cuentan con elementos del género Hack n' Slash y cuyas mecánicas han permanecido por varias generaciones de juegos, desarrollando, en su mayoría, gameplays poco complejos utilizando máquinas de estado finitas bastante simples para utilizar con los NPC y que es algo que ha permanecido en varias sagas como God of War o Devil May Cry.

A pesar del estereotipo formado por muchos de estos títulos por más de una década, existen muchos juegos que destacan de entre los demás por sus mecánicas más elaboradas, NPCs más complejos y combates relativamente más justos, que hacen que el jugador tenga qué elegir una estrategia de combate ante varios enemigos en lugar de simplemente presionar un botón repetidamente, logrando golpear incluso a varios enemigos con un solo tipo de ataque. Algunos de los representantes recientes más icónicos del género Hack & Slash son la saga de Assassin's Creed (Ubisoft – 2007) [33] y la de Batman Arkham (Rocksteady Studios – 2009) [34].

Los juegos de Batman son los que más resaltan con un sistema de combate cuerpo a cuerpo bastante fluido y llamativo en el que el jugador se debe enfrentar a más de 1 oponente a la vez (incluso más de 8) y en el que la clave es conectar correctamente movimientos en serie con cierto nivel de sincronía que permite no recibir ningún golpe de los enemigos durante un combate; sistema que incluso no alcanzaban a dominar por completo los jugadores más experimentados [35]; aún con un sistema tan vistoso de combate, el jugador solo utiliza un botón para atacar y que automáticamente elige de entre un conjunto de movimientos posibles en función de la posición de los enemigos y su distancia al jugador, existe el botón de bloqueo y algunos otras acciones que se combinan con las acciones ya mencionadas, lo que hace que la dificultad en el combate sea principalmente la sincronía para presionar los botones en el momento indicado. El entorno de combate era de tres dimensiones en el que el jugador podía desplazarse incluso aprovechando elementos de su entorno para ocultarse o incluso obtener ventajas en el combate contra los enemigos.

Por otro lado, Assassin's creed, comenzó como una franquicia nueva para Ubisoft que, además de tener un guión muy llamativo y hasta cierto punto históricamente bien documentado, ofrecía al jugador un sistema de combate en el que los enemigos se acumulaban en números grandes y de los que el jugador debía encargarse evitando morir en el intento. El combate solía ser muy difícil en concentraciones de 5 o más enemigos, debido a que más de un NPC podía atacar al jugador a la vez, mientras que el jugador sólo podía defenderse de uno de los enemigos, mas ésto era cuestión de un sistema de combate mal diseñado o sin pulir, pues en las siguientes iteraciones del juego el combate se vio más organizado pero hizo que el sistema se volviera monótono, debido a que ahora los NPCs se turnaban para atacar una vez teniendo al jugador rodeado, lo que hacía que los contraataques fueran

fáciles y solo era cuestión de esperar a los enemigos uno a uno. Tras varias entregas posteriores puede observarse que el sistema de combate mejoró en cuestiones estéticas agregando gran cantidad de animaciones como en Assassin's Creed III pero facilitando y haciendo monótono el juego para el jugador experimentado que buscaba algo de mayor dificultad [36] pero que fue mejorado en Assassin's Creed IV: Black Flag [37]. La saga de Assassin's Creed tiene como principal característica de su combate el uso de múltiples armas que pueden dar una ventaja táctica dependiendo de la situación, que van desde espadas, dagas y cuchillos arrojadizos hasta bombas de humo y dardos venenosos, que, en combinación con las acrobacias que realiza el personaje, forman parte del entorno 3D que su sistema de combate ofrece al jugador.

En resumen, podemos apreciar que tanto el Beat'em up como el Hack n' Slash, han evolucionado en contenido y complejidad en sus controles, pero no ha habido un cambio sustancial a nivel de NPCs y, actualmente, se siguen utilizando mecánicas y métodos que han cambiado poco desde los orígenes de ambos géneros.

Es muy frecuente que el único que aprende en el videojuego, mientras pasa a través de los distintos niveles, es el jugador, mientras que el NPC sigue rutinas fijas basadas en parámetros discretos que solo toman información del jugador a través del motor de juegos en lugar de tomarla de la misma manera en que el jugador lo hace haciendo un desbalance según el método aplicado.

Lo que este trabajo busca es evitar la monotonía del género Hack n' Slash sin necesidad de complejas operaciones y métodos que requieren gran capacidad de cálculo, todo esto a través de la información de conducta que aporta el jugador y que obligará al mismo evitar la monotonía en su manera de jugar, lo que le dará más valor a la experiencia de juego y que además ajustará las condiciones del entorno a la capacidad del jugador.

A grandes rasgos, la mecánica de un juego Hack & Slash gira en torno a cantidades de puntos dependiendo del medidor utilizado, que normalmente se divide en Health Points, Estamina y Mana con un tope máximo y con ciertas características cada una.

- Los **Health Points (HP)**, también conocidos como "Cantidad de Vida" o "Cantidad de Sangre", representan a lo que le resta de vida en puntos a un individuo en particular; la cantidad de vida se ve disminuida tras un ataque recibido por un contrincante sin haberlo evitado o haberse defendido, y la baja dependerá de la intensidad del ataque que fue recibido; se puede incrementar o recuperar la cantidad de puntos de vida utilizando ciertos artículos u objetos dentro del juego, aunque muchos de los juegos de hoy en día solo requieren que cierto individuo deje de recibir daño para comenzar a regenerar la cantidad de puntos de vida.
- La **Estamina**, Resistencia o Puntos de resistencia, representa la cantidad de resistencia física restante que es incluso utilizada en el lenguaje de los atletas de alto rendimiento, y en los juegos normalmente expresada en proporción (%); la estamina se gasta al realizar actividad física que requiera "más esfuerzo", como correr, saltar, evadir y en algunas ocasiones el resultado del choque de dos armas en combate; para recuperar la estamina solo hay que dejar de realizar la tarea que representaba gasto de estamina y suele regenerarse automáticamente con el tiempo.
- El **Mana** o "Magia", representa la cantidad de energía a disposición del jugador que le permite

realizar ciertos tipos de conjuros y hechizos a favor de sí mismo para su defensa o en contra de los enemigos de manera ofensiva. La cantidad de magia puede recuperarse con el tiempo como la estamina pero a un ritmo muy inferior o utilizando ciertos elementos a los que tiene acceso el jugador para una recuperación más rápida como en el caso de los puntos de salud.

Los movimientos pueden clasificarse en Ataques, Defensas y Evasiones, de la cuales se pueden subclasificar por tipo en base a la intensidad de cada movimiento, la estamina que requiere para realizarse, del daño causado o el daño evitado, que en ocasiones suele combinar más de un atributo a la vez dependiendo del caso. Tanto para el jugador como para el NPC, se tiene igualdad de acciones y movimientos disponibles con sus respectivos atributos.

Dada la estructura de un videojuego a nivel funcional, es posible representarlo con términos estadísticos, pues puede representarse la incertidumbre de la siguiente acción del jugador como una probabilidad; esto ayuda a hacer un análisis probabilístico detallado que puede involucrar varios factores en conjunto como una probabilidad condicionada o como una sucesión de eventos aleatorios. Gran parte de la información que puede proporcionar el jugador no se toma en cuenta y se desperdicia una buena fuente para análisis y encontrar una manera de contrarrestar la conducta del jugador utilizando encuentros anteriores como base de conocimientos.

Para obtener información del jugador, se puede dividir el juego en periodos de tiempo como unidad básica de medida, ésto para poder identificar ciertos comportamientos basados en duración de ciertas acciones; un ejemplo de esto, es el estado "Idle", correspondiente al estado de "espera" o "de ocio" por parte del jugador y en el cuál no se está realizando ninguna acción, que involucraría cierta cantidad de unidades de tiempo para ser validado como estado Idle. Debido a que la cantidad de movimientos está controlada y es finita, como característica de los juegos Hack n' Slash, es relativamente sencillo identificar el tipo de movimiento que se está ejecutando.

Dicho lo anterior, se puede aplicar en juego una división temporal basada en una unidad t equivalente a 0.2 segundos (p.e. $8t=1.6$ segundos) para medir la duración de cada tipo de ataque y defensa. El Anexo B contiene todos los movimientos contemplados, con su definición y justificación.

Se establecen 3 tipos de ataque: Fuerte, Medio y Débil:

Tipo	t Ataque	t Recuperación	Total
Fuerte	8	5	13
Medio	5	3	8
Débil	3	2	5

Tabla 3.1 Especificación de los tipos de ataque con su correspondientes características.

Se define la cantidad de unidades t requerida para llevar a cabo el ataque completo, así como la cantidad de unidades t para la recuperación después de llevar a cabo el ataque y realizar la siguiente acción.

Se establecen también 3 tipos de defensa: Fuerte, Débil y Bash (golpe con el escudo):

Tipo	t respuesta	A. Fuerte	A. Medio	A. Débil
Fuerte	5	3	3	3
Débil	1	7	3	2
Bash	4	10	3	2
Evasión	1	5	5	5

Tabla 3.2 Especificación de los tipos de defensa y evasión con su correspondientes características.

Se define la cantidad de unidades t necesarias para realizar cada tipo de defensa (t respuesta), así como la cantidad de unidades t necesarias para recuperarse después de cada tipo de ataque recibido según la defensa (t Recuperación).

Los ataques tienen establecidos los siguientes parámetros:

Daño: Es la cantidad de daño hecho al oponente en el tiempo T. Conforme más alto sea el tiempo T se acercará al daño máximo de daño.

Stamina de Ataque: Es la stamina gastada acumulada según el tiempo T del ataque.

Stamina de Bloqueo: Es la stamina total gastada en caso de que el ataque sea bloqueado en el tiempo T.

Bash Penalty: Es la stamina extra gastada en caso de que el oponente realice un Bash mientras se está realizando el ataque. El porcentaje descrito se suma a la stamina de Ataque en el tiempo T, siendo el 100% el total de Stamina que se gasta al llegar al último tiempo T del ataque con el 100% de daño.

Counter Penalty: Es la stamina extra gastada en caso de que el oponente realice un contraataque con éxito antes de que el daño llegue a 100%.

Efectivo: Establece el tiempo T en el que el ataque se considerará como bueno para su posterior peso en la macro que se está ejecutando.

Capítulo 4

ESTIMACIÓN CON EL USO DE PROCESOS ESTOCÁSTICOS.

Ya establecidos los conceptos esenciales en materia de estadística, machine learning, videojuegos y las características del género Hack & Slash, podemos ahora unir las partes para dar forma a la propuesta que ofrece este trabajo de tesis, y que está dividido en dos grandes apartados: (1) El pronóstico del Estatus del jugador y (2) el Pronóstico de los movimientos del jugador.

4.1 Pronóstico del Estatus del Jugador.

Se define como *Estatus del Jugador* al conjunto de atributos del mismo, representado por una variable aleatoria, en cuestiones de Salud (HP) y Estamina (S) basados en la lista de acciones establecida según el tipo de movimiento: Ataque (A) y Defensa(D). El objetivo es encontrar la probabilidad de que el jugador se encuentre en una situación determinada (el espacio muestral representado por “Alto” y “Bajo” de cada variable) según el movimiento en el estado actual.

El Estatus del Jugador es una variable aleatoria debido a que para un NPC, los movimientos y decisiones del jugador son aleatorios, esto es, que no se sabe con certeza lo que el jugador va a llevar a cabo.

Por medio de la probabilidad condicionada podemos calcular la probabilidad de que el Estatus del jugador se encuentre en cierta condición dados los movimientos que está ejecutando. Se dividirá principalmente en dos:

- Cantidad de Estamina Restante: Basada en los tipos de ataque y defensa realizados del jugador.
- Cantidad de Salud Restante: Basada en la cantidad de ocasiones en que el jugador defendió o evadió un ataque.

La otra variable aleatoria involucrada en la probabilidad condicionada es el tipo de ataque que puede tener como espacio muestral. Los distintos tipos de ataque definidos en el Anexo A.

Comenzando por la cantidad de estamina, por medio del Teorema de la Probabilidad Total se define la probabilidad de un Ataque Fuerte (AF) :

$$P(AF) = P(AF/SB)P(SB) + P(AF/SA)P(SA)$$

Donde SB representa a la estamina baja. Tomando en cuenta lo anterior, podemos entonces definir por medio del teorema de Bayes:

$$P(SB/AF) = \frac{P(AF/SB)P(SB)}{P(AF/SB)P(SB) + P(AF/SA)P(SA)} \quad (1)$$

$$P(SA/AF) = \frac{P(AF/SA)P(SA)}{P(AF/SB)P(SB) + P(AF/SA)P(SA)} \quad (2)$$

En el caso de (1) se interpreta como La probabilidad de que la Estamina sea baja, dado que hubo un ataque fuerte. Mientras que (2) se interpreta como la probabilidad de que la Estamina sea Alta dado que hubo un ataque fuerte.

El umbral de la estamina se ha establecido en 45%, es decir, cuando la estamina es menor o igual al 45% la estamina se define como baja.

4.2 Pronóstico de los movimientos del jugador.

Otro objetivo es predecir los movimientos del jugador (A, D, E)..... en este ejemplo solo 3 ataques. En este apartado se desea pronosticar el movimiento en el instante t_{n+1} a partir del movimiento del instante t_n . Según los tipos de ataques básicos definidos en el Anexo D, se consideran los posibles estados como Ataque Débil (AD), Ataque Medio (AM) y Ataque Fuerte (AF) definiendo la matriz de transición:

		Tiempo $n+1$		
		AF	AM	AD
Tiempo n	AF	$p_{AF,AF}$	$p_{AF,AM}(n)$	$p_{AF,AD}(n)$
	AM	$p_{AM,AF}(n)$	$p_{AM,AM}$	$p_{AM,AD}(n)$
	AD	$p_{AD,AF}(n)$	$p_{AD,AM}(n)$	$p_{AD,AD}$

Tabla 4.1 Matriz de transición para los tipos de ataque definidos.

La matriz de transiciones genera el siguiente diagrama de estados:

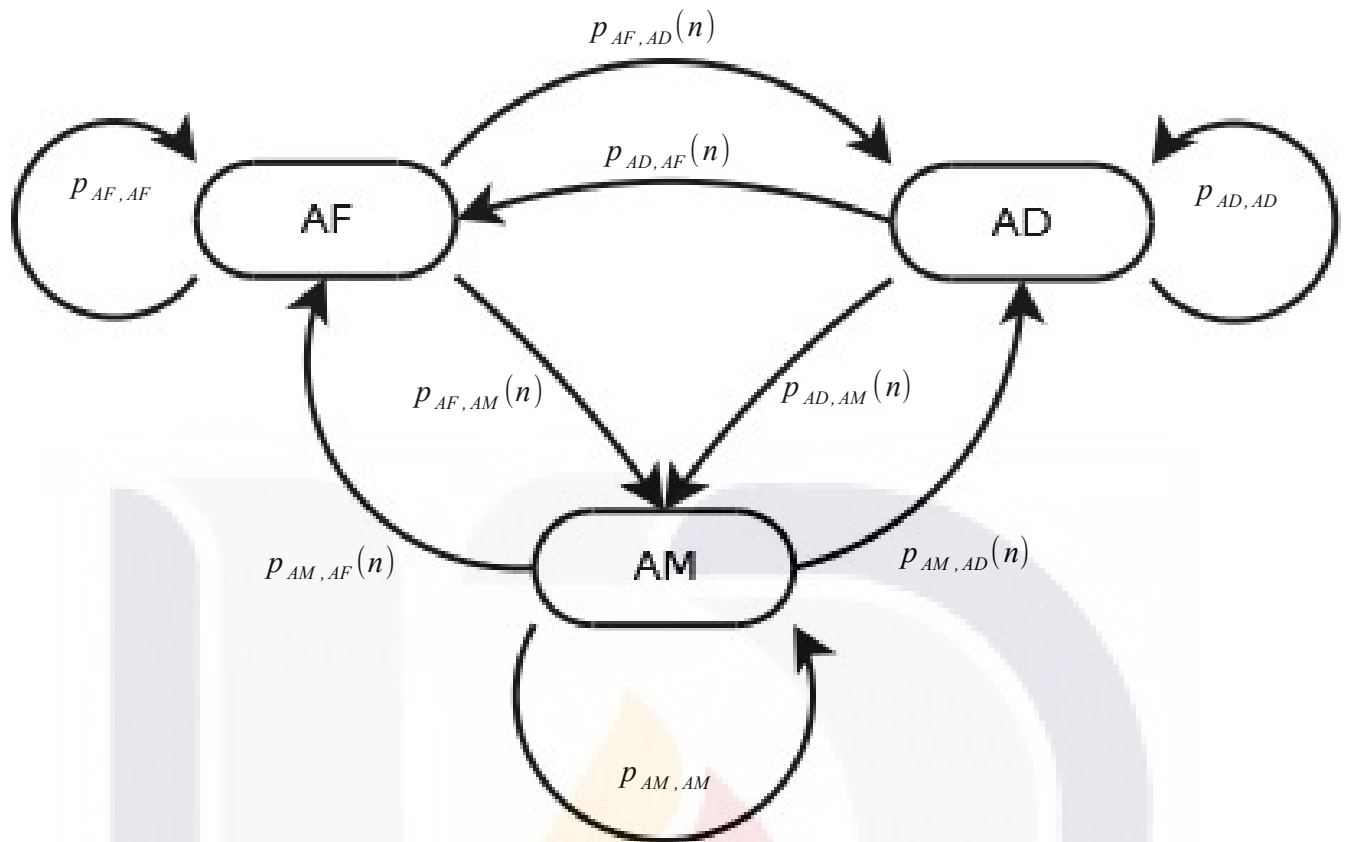


Figura 4.1 Representación de la matriz de transición como un diagrama de estados.

todo esto basado en la transición del tiempo n al tiempo $n+1$ que se representa como una Cadena de Márkov.

Se requiere analizar la probabilidad de ir de un estado a otro después de k unidades de tiempo que serán definidas dentro de los experimentos como la cantidad de acciones que contendrá una macro para definir un combo.

Las probabilidades se calculan en base a los registros de la sucesión de ataques del jugador. Es necesario probar con distintos valores de k para la comparación y evaluación de resultados durante las simulaciones.



Capítulo 5

EXPERIMENTOS.

5.1 Diseño de Experimentos.

Una vez definidos todos los movimientos posibles se almacenan, en una matriz de transición como en la *Tabla 3.1*, la cantidad de ocasiones en que se realizó el movimiento X después del movimiento Y y, una vez que se tiene el total de ocasiones en las que ocurre una transición entre X y Y , se calcula la probabilidad con respecto al total de transiciones en la matriz utilizando el Teorema de Bayes en tiempo de ejecución, lo que permite actualizar las probabilidades de la matriz inmediatamente después de almacenar una transición vista en el jugador durante el combate.

Para crear y adecuar un modelo básico, del cuál se pueda comprobar su efectividad, se utilizó el lenguaje de programación Python [38] debido a su sencillez, portabilidad y el rendimiento que ofrece, debido a sus propiedades como Lenguaje Interpretado. El lenguaje de Scripting LUA, que es similar a Python en la arquitectura de funcionamiento y su soporte multiplataforma, es uno de los lenguajes más utilizados en materia de Scripting en Inteligencia Artificial en juegos [39][40]. Se decidió utilizar Python en esta investigación debido a la flexibilidad que ofrece en cuestiones de cálculos matemáticos gracias a librerías como Scipy [41], que extienden la funcionalidad para administrar grandes cantidades de memoria y manejo de arreglos de proporciones y dimensiones que se frecuentan en el cómputo científico, lo que permite escalar el modelo propuesto en este trabajo a un nivel que da lugar a un análisis más complejo y distribuible en distintos hilos para procesamiento paralelo con la librería Parallel Python [42].

Se establecen tres tipos básicos de ataque con sus respectivos consumos de estamina con respecto al total de la misma:

- D: Débil (5% de Estamina)
- M: Medio (15% de Estamina)
- F: Fuerte (25% de Estamina)

Debido a que el procesamiento de cada cadena de movimientos es secuencial, es necesario añadir un indicador que nos permita identificar tanto el inicio como el fin de cada uno de los combos, por lo que agregamos un estado "Idle" (o de ocio) a partir del cual se inician todos los combos y con el cuál se culminan.

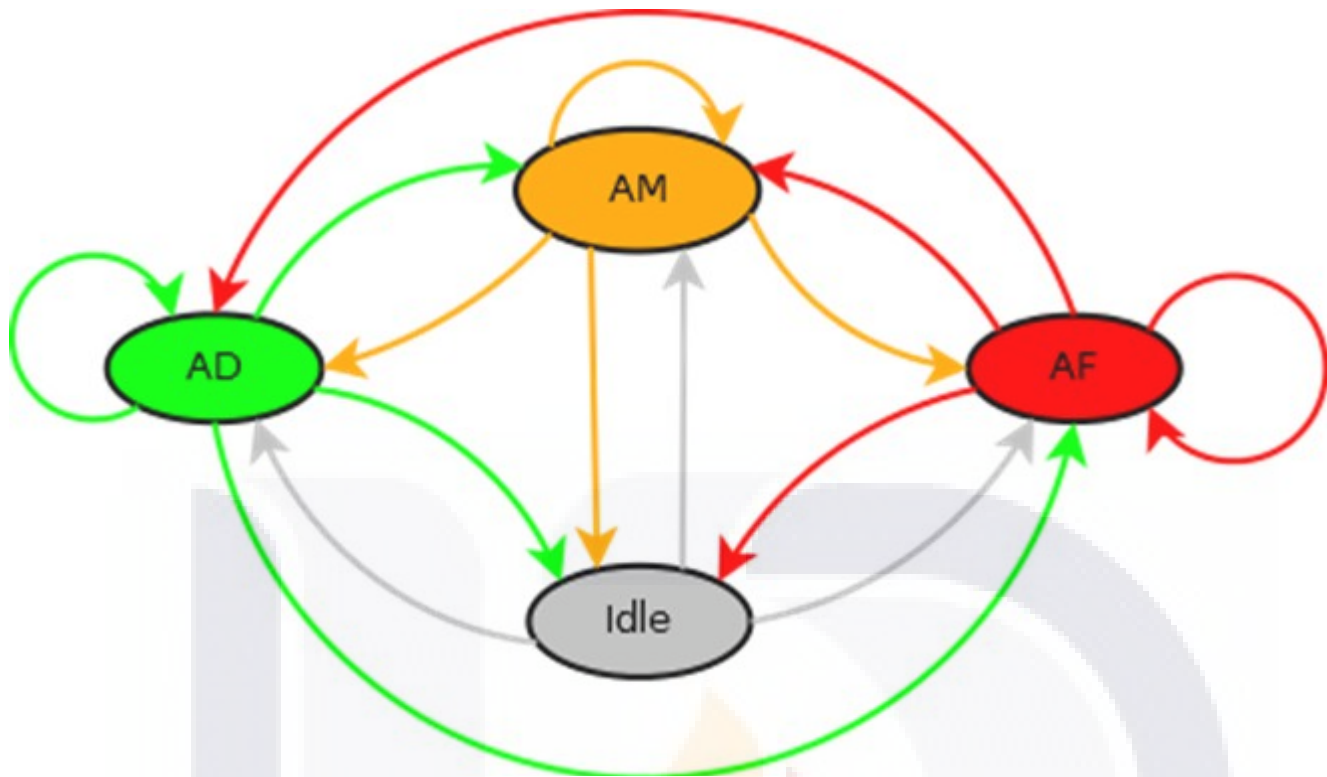


Figura 5.1 Diagrama de estados final para la fase de experimentos.

Se ha definido una serie de Combos para capturar las transiciones. Cada uno de los combos lleva implícito al inicio y al final se incluye un estado “Idle” como el mostrado en la *Figura 3.2*, ésto para marcar el inicio y final de cada una de las combinaciones de movimientos, se ha obviado de las listas por cuestiones de redundancia. Por lo tanto, la matriz de transición quedaría finalmente como se muestra en la *Figura 3.4*, agregando el Estado Idle como una transición más.

Como puede observarse, la transición del Estado Idle al Estado Idle se iguala a cero, ésto debido a que el estado Idle siempre será el inicio o final de un combo, por lo que nunca existirá una transición al mismo estado tal como puede apreciarse en la *Figura 3.2*.

Se optó por dos maneras de tener una matriz de transición, la primera fue guardar una matriz en un archivo CSV y la otra fue obtener una matriz al vuelo definiendo cuántos combos se utilizarían para llenar la matriz de transición (*Tabla 3.4*); ambas opciones pueden apreciarse de manera general en el *Anexo C*.

		Tiempo $n+1$			
		AF	AM	AD	I
Tiempo n	A F	$p_{AF,AF}$	$p_{AF,AM}(n)$	$p_{AF,AD}(n)$	$p_{AF,I}(n)$
	A M	$p_{AM,AF}(n)$	$p_{AM,AM}$	$p_{AM,AD}(n)$	$p_{AM,I}(n)$
	A D	$p_{AD,AF}(n)$	$p_{AD,AM}(n)$	$p_{AD,AD}$	$p_{AD,I}(n)$
	I	$p_{I,AF}(n)$	$p_{I,AM}(n)$	$p_{I,AD}(n)$	$p_{I,I}=0$

Tabla 5.1 Matriz de transición incluyendo el estado Idle.

La matriz de transición modificará la probabilidad de transición según se va capturando cada uno de los movimientos por separado en cada iteración tomando en cuenta el total de movimientos capturados en total.

Este diseño de captura cuenta con un exploit, ésto debido a que si se optara por utilizar sólo un tipo de movimiento para llenar la matriz de transición con miles de movimientos, la probabilidad de transición de la matriz en esa transición en particular sería muy alta:

	AF	AM	AD	I
AF	0	0	0	0
AM	0	0	0	0
AD	0	0	0.8	0.1
I	0	0	0.1	0

Tabla 5.2 Matriz de transición desbalanceada.

Si el jugador cambiara el patrón de ataque después de una cantidad relativamente grande de ocasiones de utilizar un solo tipo de ataque, la estimación continuaría tendiendo hacia la transición con mayor probabilidad de ocurrencia (*Tabla 3.6*).

	AF	AM	AD	I
AF	0	0	0	0
AM	0	0	0	0
AD	0	0	0.75	0.1
I	0.05	0	0.1	0

Tabla 5.3 Matriz de transición desbalanceada actualizada.

Estadísticamente, entre mayor sea el total de movimientos, los cambios de tendencia serán ínfimos frente a la proporción que representa el patrón anterior. Si se tiene un número máximo de movimientos registrados muy pequeño, el modelo no tendría sentido, ésto debido a que se desecha la información muy rápidamente y no puede obtenerse un patrón de comportamiento que resulte significativo como para lograr adaptarse. La solución a este problema es obtener un número de movimientos balanceado que permita adaptarse en la menor cantidad de iteraciones posibles, manteniendo un historial de combate que sea lo suficientemente representativo del patrón de movimientos del jugador.

Se decidió un máximo de valores capturados t_{max} que toma los valores 20, 50, 100, 200, 500, 1000 y 2000 movimientos almacenados en la matriz de transición que se van desechando en tandas de la mitad para lograr observar su adaptación a un cambio radical en las combinaciones realizadas, que en aplicaciones prácticas podría incluso soportar un cambio de jugador en medio de un combate. Puede analizarse más a detalle el algoritmo de depurado de la base de conocimientos en el *Anexo D*.

Tras el procesamiento de una transición, se guarda la probabilidad de cierto movimiento dado el nivel de estamina, que ayuda a estimar un evento en base al nivel de estamina para complementar el ya conocido por parte de las transiciones entre los tipos de ataque.

Se ha definido dos grupos de 6 y 20 combos, denominados respectivamente X y Y, como se muestran en la *Tabla 3.7* y la *Tabla 3.8* con secuencias similares para llenar la base de conocimientos de la matriz de transiciones. La similitud en transiciones fue creada para ligeras variaciones y probar el modelo ante patrones semejantes.

Combo 1	M	M	M		
Combo 2	D	M	D		
Combo 3	M	F	F		
Combo 4	D	D	M	M	
Combo 5	M	M	F	D	
Combo 6	D	M	F	M	M

Tabla 5.4 Estructura de combos con seis variantes (X).

Combo 1	M	M	M	M	M
Combo 2	M	M	F	D	
Combo 3	M	D	M		
Combo 4	M	F	M		
Combo 5	D	M	D		
Combo 6	D	M	F	M	M
Combo 7	D	D	M	M	
Combo 8	D	D	F		
Combo 9	M	F	F		
Combo 10	M	F	D	D	D
Combo 11	M	M	F		
Combo 12	M	D	F		
Combo 13	D	D	M	M	
Combo 14	D	D	M		
Combo 15	D	M	M	D	D
Combo 16	D	F	M		
Combo 17	F	D	D	D	D
Combo 18	F	D	M		
Combo 19	F	D	M	D	
Combo 20	F	M	M	D	

Tabla 5.5 Estructura de combos con 20 variantes (Y).

Como puede apreciarse, cada tabla tiene similitudes en transiciones entre sus combos, pero en la tabla con 20 variantes de combos existe un cambio significativo en las transiciones, sobre todo en las transiciones iniciales y finales, pues en el grupo de combos de 6 no existe ningún combo que inicie con un golpe del tipo “Fuerte”, por mencionar alguna.

5.2 Condiciones de Inicialización y Pruebas.

Se inicializa para cada prueba la Matriz de transición con 500 combos aleatorios del grupo X, que generan entre 1500 y 2500 movimientos, garantizando que se realizará en al menos una ocasión el depurado de movimientos según el valor t_{max} . La matriz vuelve a inicializarse en cada ocasión en que cambia el valor t_{max} y el grupo de combos a evaluar, dando lugar a 14 pruebas distintas.

Se ejecuta el procesamiento de ambos grupos de combos 50 veces en cada una de las pruebas, con un total de combos capturados de 300 para el grupo X y 1000 para el grupo Y en la Matriz anteriormente inicializada. g

Para comprobar la efectividad del modelo propuesto, se contabilizan las ocasiones en que la estimación de una transición fue correcta al comparar los valores estimados por la matriz de transición con respecto al combo que se está analizando. Todos los combos en ambos grupos incluyen el estado Idle al final, lo que actualiza la probabilidad de transición de todos los movimientos hacia el estado Idle, que representa el último movimiento en cada combo.

5.3 Resultados.

Los resultados de las ejecuciones pueden verse a detalle en la *Tabla 3.9*, incluyendo todas las combinaciones en tamaño máximo de movimientos capturados y el grupo de combos utilizado.

Prueba	Promedio	Mediana	Desviación	Min	Max
20 Mvs – 6 Combos	0.6285427123	0.6280922039	0.0060608964	0.6071428571	0.6607142857
20 Mvs – 20 Combos	0.7063571523	0.7052631579	0.0045030519	0.7052631579	0.7368421053
50 Mvs – 6 Combos	0.6139202126	0.6111678005	0.0077441026	0.6071428571	0.6428571429
50 Mvs – 20 Combos	0.7060683072	0.7052631579	0.0021411898	0.7052631579	0.7157894737
100 Mvs – 6 Combos	0.5671552622	0.5678806005	0.0061347302	0.5357142857	0.5773809524
100 Mvs – 20 Combos	0.6171158097	0.6172280702	0.0046846724	0.5894736842	0.6280701754
200 Mvs – 6 Combos	0.5504761542	0.5531908001	0.011381999	0.5	0.5616883117
200 Mvs – 20 Combos	0.5102235524	0.5092838196	0.005702753	0.4842105263	0.5245614035
500 Mvs – 6 Combos	0.5246186362	0.5385129279	0.0335879046	0.4285714286	0.5496031746
500 Mvs – 20 Combos	0.4393939992	0.4394736842	0.0034185492	0.4210526316	0.4456140351
1000 Mvs – 6 Combos	0.5137306073	0.5265916149	0.0247935739	0.4642857143	0.5316558442
1000 Mvs – 20 Combos	0.4209780794	0.417854251	0.0105148197	0.4147368421	0.4631578947
2000 Mvs – 6 Combos	0.5466107091	0.5476098901	0.0095454764	0.5357142857	0.56
2000 Mvs – 20 Combos	0.4001921581	0.3968859649	0.0115071602	0.3939849624	0.4631578947

Tabla 5.6 Resultados de las pruebas según el grupo de Combos y el tamaño máximo de movimientos almacenados.

5.4 Discusión de Resultados.

Puede observarse en los resultados una diferencia en la probabilidad promedio entre el grupo X y el grupo Y con el mismo tamaño de matriz, esto representa el éxito en la adaptación del modelo en lo que respecta al grupo de combos diferentes, puesto que la prueba con el grupo X implica la adaptación al mismo grupo original con el que se inicializó la matriz, mientras que el grupo Y pone a prueba la adaptación de una inicialización a un grupo de transiciones contrastante.

Cabe destacar que la probabilidad de éxito en la estimación en las pruebas de 20, 50 y 100 movimientos para el grupo X es menor a la probabilidad de éxito en el grupo Y , lo que indica que el grupo X cuenta con cierto balance en sus transiciones, mientras que la diferencia en las transiciones es más marcada y más identificable en el grupo Y , logrando incluso una adaptación más exitosa en comparación con el grupo X .

En las pruebas de 200, 500, 1000 y 2000 puede notarse que la probabilidad de éxito es superior en el grupo X si se compara con el grupo Y dado que la base de conocimientos tiene una tendencia muy marcada a las transiciones del grupo X , identificado como un exploit en el diseño de experimentos en el apartado 3.4 del presente trabajo.

Si se compara la Desviación Estandar de cada una de las pruebas, el valor menor representa a la prueba de 20 combos con 50 movimientos capturados, con un valor de 0.0021.

TESIS TESIS TESIS TESIS TESIS

Por lo tanto, el máximo óptimo de movimientos capturados para cuatro estados distintos (AF, AM, AD, I) es de 50 mostrando una probabilidad de estimación exitosa aproximada del 70%, dicho de otra manera, el modelo ofrece una capacidad de adaptación en el 70% de las ocasiones; esta proporción es más que aceptable, debido a que con el 50% de las ocasiones se obtendrá un balance entre la cantidad de estimaciones exitosas, que es la tendencia ideal como puede verse en artículos relacionados de otros autores [24] que están relacionados en el género del juego objetivo.

Si la proporción de adaptación tendiera hacia el 100% de éxito, el juego sería imposible de vencer, provocando que el jugador termine por dejar el juego al romper el balance de la dificultad con respecto al jugador mismo.

Las pruebas realizadas demuestran que es posible definir un nivel de dificultad “discreto” basado en el tamaño de la base de conocimientos, donde la dificultad será menor si la cantidad de movimientos almacenados es menor.

En resumen, fue logrado el uso de procesos estocásticos y la probabilidad condicional que ofrece el Teorema de Bayes y las cadenas de Markov para generar un método de Aprendizaje Máquina no supervisado que se adaptará al jugador según la base de conocimientos y la cantidad de enfrentamientos con el jugador.

Se lograron abstraer con éxito los elementos de Gameplay de un Hack n' Slash para poder estimar la condición en la que se encuentra el jugador basado en los movimientos realizados y que ayudan a mejorar la base de conocimientos para pronosticar los movimientos posteriores utilizando la probabilidad condicionada entre la estamina y los movimientos posibles.

Gracias a las cadenas de Márkov, fue posible definir una secuencia de movimientos de combate del jugador de manera dinámica dentro del entorno controlado de un Hack n' Slash, ayudando a mejorar la base de conocimientos de una manera rápida y sin utilizar grandes recursos para cálculo y ejecución.



CONCLUSIONES

Es posible identificar patrones de comportamiento en un jugador en el ambiente controlado que ofrece un Hack & Slash utilizando los procesos estocásticos como herramienta de evaluación. Tras realizar las pruebas con distintos tamaños de la base de conocimientos y encontrar el tamaño máximo óptimo, podemos observar que es posible lograr un nivel de adaptación aceptable según el patrón de comportamiento observado en un jugador. Con estos resultados el desarrollador puede adoptar los parámetros que considere óptimos para su aplicación, pues si se desea una Inteligencia Artificial adaptativa que mantenga un balance con respecto al jugador, puede optarse por un tamaño de base de conocimientos que genere un 50% de probabilidad de estimación exitosa; pueden incluso tomarse distintos tamaños de la base de conocimientos para definir distintos niveles de dificultad, siendo el de mayor probabilidad de éxito el de dificultad más alta.

Como se puede observar, sin necesidad de el uso de cheats y con una cantidad de información equivalente entre el jugador y el NPC, se puede lograr un comportamiento inteligente creíble en los NPCs que aporten un reto al jugador. La tarea es obligar al jugador a ser más dinámico en la selección de movimientos para vencer al NPC que implemente el modelo propuesto, haciendo que la rejugabilidad sea más rica y logrando que el jugador tenga una experiencia distinta en cada ocasión que vuelva a iniciar un juego desde el inicio.

El modelo propuesto puede aplicarse no solo a los juegos del género Hack n' Slash sino también a aquellos juegos donde pueda definirse un ambiente controlado y acotado por un conjunto de parametros finitos e identificables, como puede ocurrir en los juegos de estrategia por turnos, los juegos de pelea, los RPG con sistemas de combate por turnos, y otros géneros que puedan adaptarse al modelo propuesto; representa una herramienta que puede adaptarse a las necesidades del desarrollador, el cuál decidirá cuáles variables son las que entrarán en juego para el análisis y que generarán resultados para la toma de decisiones.

Debido a la baja complejidad de cálculo requerida para el método de adaptación, el actual modelo puede ser aplicado a dispositivos con bajos recursos de hardware como los dispositivos móviles o plataformas que establezcan una prioridad baja a los métodos de optimización de Inteligencia Artificial como las consolas de videojuegos. Si se cuenta con recursos más altos en el equipo que ejecutará el juego, se pueden agregar, bajo el mismo concepto, otras variables aleatorias para relacionarse entre sí utilizando la probabilidad condicionada y la probabilidad de transición según sea el caso.

Se desea realizar a futuro una herramienta más robusta que tenga la capacidad de procesar una mayor cantidad de variables aleatorias y extrapolarlo a juegos más complicados de analizar como los del género shooter, que muestran un nivel de complejidad superior debido a que usualmente el jugador se enfrenta a más de un NPC a la vez con distintas rutas y a una distancia que en ocasiones hacen que alguno de los involucrados no pueda divisar a su contrincante rápidamente. La cantidad de elementos y herramientas que el jugador tiene a disposición en un juego shooter, hacen que sea necesario un análisis más profundo de las situaciones que, si bien incrementan su nivel de complejidad, pueden seguir siendo acotados en espacios muestrales mucho mayores pero que siguen

siendo finitos.



GLOSARIO

Beat 'em up

También conocido como brawler, es un género de videojuegos que consiste en el combate melee entre el protagonista, controlado por el jugador, y una indefinida y enorme cantidad de enemigos con habilidades inferiores. Normalmente este tipo de juegos tomaban lugar en asentamientos urbanos e incluían pelear contra el crimen o argumentos basados en venganzas como Double Dragon, o incluyendo algunos con trama histórica o temas de fantasía como Golden Axe, Captain America and the Avengers o Teenage Mutant Ninja Turtles que fueron populares en las máquinas de Arcadia.

Cheat:

Del inglés “*engaño*”, hace referencia al uso de trampas en el ámbito de los videojuegos, utilizando herramientas que proporcionan información adicional a la que está previamente establecida por la mecánica del juego mismo. Algunos ejemplos de Cheats incluyen el ver a otros individuos y dispararles a través de las paredes (Wall-hack), puntería automática (Aimbot), inhibición de físicas que permiten el movimiento libre más allá de los límites físicos previamente definidos del escenario por parte del diseñador del escenario o mapa (No-Clipping), por mencionar algunos. En términos de desarrollo de videojuegos, el término Cheat se refiere al uso de ventajas sobre el jugador gracias a la información que proporciona el motor de juego durante la ejecución de un videojuego, con lo que es posible conocer la posición exacta del jugador, el arma que porta junto con la cantidad de munición o maná (si aplica), la cantidad de puntos de salud restantes y sus objetos en el inventario, por mencionar algunos, que permiten, bajo mecánicas rudimentarias, obtener rápidamente una ventaja sobre el jugador según el nivel de dificultad elegido. Debido a las limitantes de hardware en las consolas, es común que los desarrolladores opten por este tipo de “ventajas” en lugar de utilizar métodos más complejos de análisis del jugador, haciendo que los NPCs tengan información perfecta y no el jugador, haciendo el juego injusto si se habla de teoría de juegos.

Combo

En videojuegos, un combo (abreviación de “combination”) es un término referente a un conjunto de acciones ejecutadas en secuencia, usualmente con limitaciones estrictas de sincronía, que brindan una ventaja o beneficio significativos. Una de las primeras mecánicas de Combo en videojuegos se encuentra en la serie de Street Fighter, en el título Street Fighter II Turbo, donde se indicaba en pantalla la cantidad de golpes conectados al contrincante. Desde entonces ha sido aplicado a gran variedad de géneros, desde juegos de tipo Puzzle y “Beat'em up” a juegos de deportes. Los combos también son utilizados como un elemento esencial de Gameplay, o utilizado meramente como un modificador de potencia de ataque o alto puntaje, o simplemente como una manera de exhibir un estilo de juego extravagante, no indispensable para una victoria o la supervivencia durante el juego.

Consola

También conocida como consola de videojuegos o videoconsola, es un sistema electrónico de entretenimiento para el hogar que ejecuta juegos electrónicos (videojuegos) contenidos en Cartuchos, Discos ópticos, discos magnéticos, tarjetas de memoria o cualquier dispositivo de almacenamiento.

Escalabilidad de la Dificultad

La escalabilidad de la dificultad es la adaptación automática de un juego para balancear el reto que éste representa para un jugador humano. Es un término que comúnmente se encuentra en textos científicos que hablan sobre Inteligencia Artificial Adaptativa y su adaptación se basa en el nivel de pericia o dominio de un jugador hacia un juego. Una de las características más particulares de este tipo de dificultad, es que se hace una adaptación en las tácticas utilizando métodos de evaluación que logren una Inteligencia Artificial que sea difícil pero que se pueda vencer, dependiendo de la capacidad de cada tipo de jugador. Se considera que un juego es divertido si representa un reto a vencer, siempre y cuando no diezme la capacidad del jugador y haga al juego parecer imposible de vencer.

Exploit

En videojuegos, es el uso de un bug o glitch, hit boxes, velocidad, entre otros, por un jugador para su ventaja de una manera que no había sido deseada por los diseñadores y programadores del juego. Se le abrevia comúnmente de manera coloquial como “sploit”. Los Exploits han sido clasificados como una forma de hacer trampa, de cualquier manera, la determinación precisa de qué es lo que se considera o no exploit puede ser controversial.

Fog of War

Del alemán *Nebel des Krieges*, se refiere a la incertidumbre en la conciencia de una situación que es experimentada por los participantes en las operaciones militares. El término busca capturar la incertidumbre con respecto a la capacidad de uno mismo, del adversario y el propósito del adversario durante un combate, operación o campaña. De manera similar, en videojuegos, principalmente en el Género RTS, se representa como una parte del escenario en la que no se tiene ninguna unidad que pueda ver de cerca y que permita al comandante ver lo que los contrincantes se encuentran haciendo, se representa normalmente en el juego como la parte sombría en un mapa, donde no es posible observar fácilmente unidades enemigas.

Gameplay

Es lo que distingue el género de un videojuego. Define las mecánicas de juego que se utilizan para que el jugador interactúe con el juego mismo. También se le conoce como mecánica de juego

Hardcore Gamer:

Referente al videojugador asiduo, caracterizado principalmente por dedicar gran cantidad de horas de tiempo a jugar videojuegos y tienen un gran conocimiento sobre los videojuegos en general desde un punto de vista objetivo; no están satisfechos con terminar un juego de manera habitual, pues buscan siempre conseguir terminarlo mientras el juego implique un reto de manera competitiva para poner a prueba sus habilidades. El Hardcore Gamer busca más un reto que el simple medio de entretenimiento y son los que más comúnmente se percatan de detalles como la calidad de la Game AI.

Hack ‘n Slash

También conocido como “Hack and Slay” se refiere a un tipo de gameplay que hace énfasis en el combate. Fue utilizado originalmente para describir un estilo de juego en juegos de rol basados en

tablero llevándolos a los MMORPG y a los videojuegos en general. El término implica específicamente el enfoque en el combate mano a mano con armas.

In-game:

Es un término utilizado para hacer alusión a los elementos dentro de un videojuego, como puede ser el escenario, los objetos, animales, enemigos o incluso la atmósfera del juego.

Inmersión

En videojuegos, se define como la medida general de entretenimiento. Implica la sensación de adentrarse en un videojuego a través de efectos visuales, de sonido e incluso situaciones que generan estrés, haciendo que el jugador mantenga el interés en el juego.

Item

En videojuegos, un item es un objeto dentro del mundo del juego mismo que puede ser obtenido por el jugador, u ocasionalmente a través de un NPC. También son conocidos como “Pick-ups” y se guardan en el “Inventario” del jugador.

Non-Player Character

Término a menudo abreviado como NPC, también conocido como “bot”, es un personaje controlado por el juego. El NPC es normalmente parte del programa y no es controlado, pero en ocasiones sí orientado, por el jugador humano. El origen del término viene de los juegos de Rol (RPG) pero se ha generalizado para hacer referencia a cualquier personaje que no sea el jugador humano, involucrando tanto a amigos como enemigos del jugador.

Rejugabilidad

Del inglés “Replayability” o “Replay Value”, es un término normalmente encontrado en combinación con los videojuegos, pero también puede ser utilizado para describir otros tipos de juegos, películas, música u obras de teatro. En Videojuegos, el término se utiliza para describir el valor de entretenimiento al jugar un juego más de una vez. Anteriormente, la rejugabilidad de un juego podía basarse en gustos individuales, pues podría involucrar la calidad de la música, los gráficos o encontrar el juego divertido. En general, engloban los ambientes dinámicos del juego, una Inteligencia Artificial retadora, una amplia variedad de tareas por completar, y un buen conjunto de atractivos que mantendrán al jugador junto al juego por una cantidad de tiempo indefinidamente extensa.

RPG (Role-Playing Game)

Es un tipo de juego en el que el jugador controla las acciones de un cierto personaje mientras éste se encuentra en el ambiente generado por el juego. Muchos juegos de Rol tienen su origen en juegos de rol de “Lápiz y Papel” como “Calabozos y Dragones”, utilizando mucha de la misma terminología, ajustes y mecánicas de juego. Maneja puntos de Experiencia (XP) en base a ciertas acciones que suele realizar el jugador, como lo es el caso del combate o cumplir ciertas misiones y tareas (Quest).

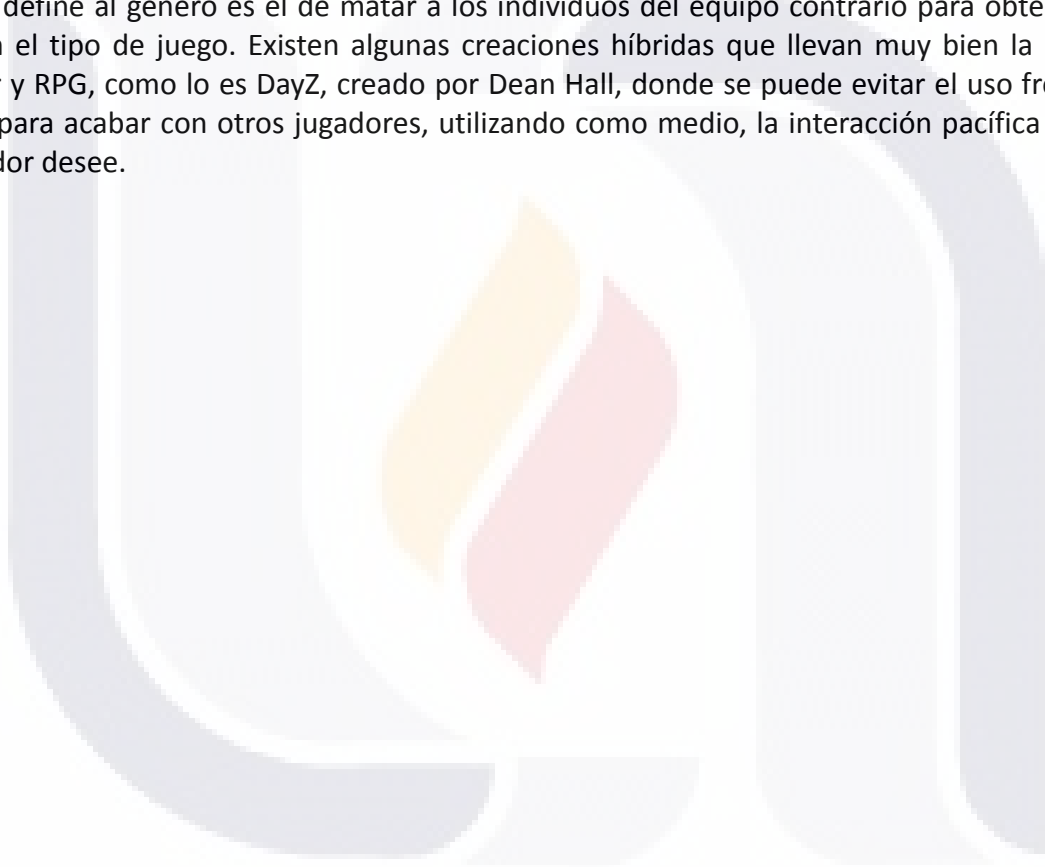
RTS (Real-Time Strategy)

Es un subgénero de el género de Estrategia en videojuegos en el que no existe un progreso basado en turnos. El jugador funge en el papel de comandante y es capaz de manejar más de un personaje,

comúnmente formando grandes grupos, que seguirán las instrucciones que el jugador haya seleccionado. En éste género, el término NPC podría considerarse erróneo si se refiere a todas las unidades que conforman el ejército enemigo, pues, si se considera igualdad de circunstancias, el NPC sería el comandante enemigo que es quien se encarga de dar las órdenes a su respectivo ejército.

Shooter

Es un sub-género de los juegos de acción, el cuál comúnmente prueba la velocidad de reacción del jugador. Incluye varios subgéneros en los que se enfoca en las acciones de un personaje utilizando algún tipo de arma. Comúnmente el arma referida es un arma de fuego o incluso alguna otra arma de largo alcance como el arco o la ballesta. El jugador normalmente tiene la vista de los eventos a través de los ojos del personaje (First-Person Shooter) o a través de una cámara que sigue al personaje, y que se sitúa a una distancia relativamente pequeña detrás del personaje (Third-Person Shooter). El estándar que define al género es el de matar a los individuos del equipo contrario para obtener una ventaja según el tipo de juego. Existen algunas creaciones híbridas que llevan muy bien la relación entre Shooter y RPG, como lo es DayZ, creado por Dean Hall, donde se puede evitar el uso frecuente de las armas para acabar con otros jugadores, utilizando como medio, la interacción pacífica u hostil según el jugador desee.

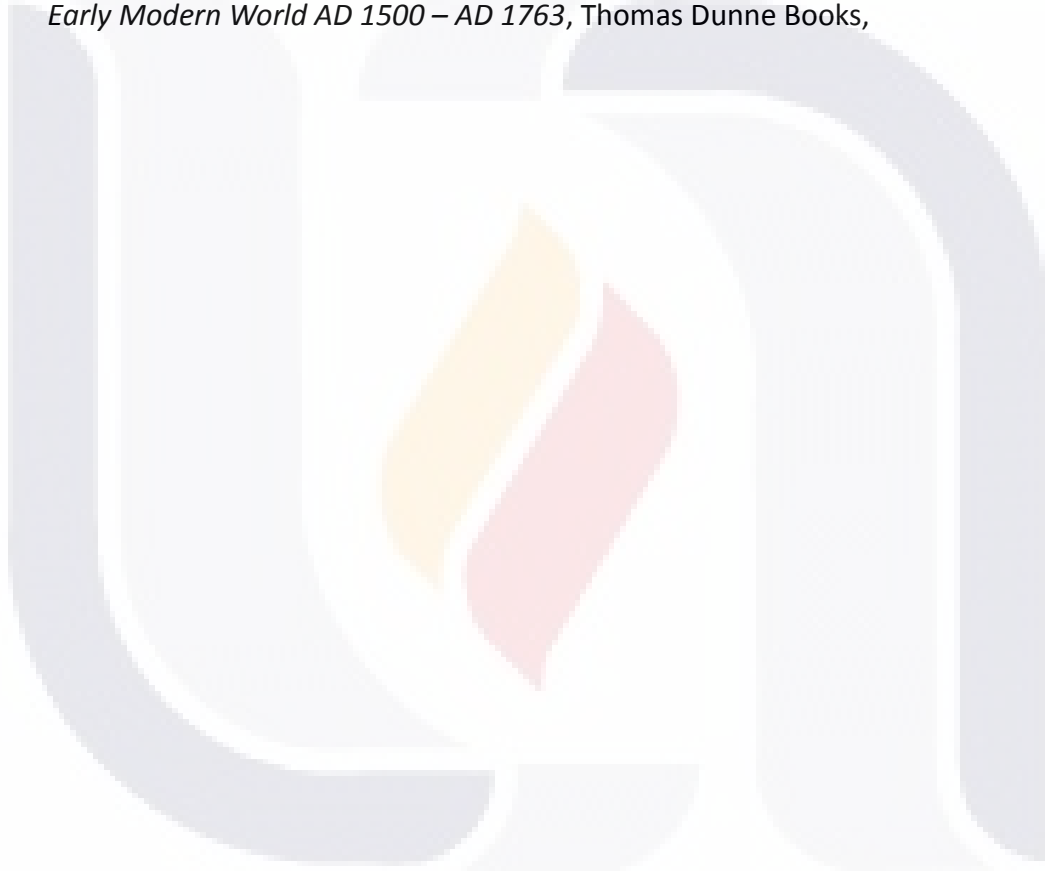


BIBLIOGRAFÍA

- [1] S. Bakkes, P. Spronk, J. Van den Herik, "Rapid Adaptation of Video Game AI", en *Proceedings of the IEEE Symposium on Computational Intelligence in Games 2008* (eds. Luigi Barone and Philip Hingston), pp. 79-86, Diciembre 2008.
- [2] T. Mitchell, *Machine Learning*, McGrawHill, 1997.
- [3] Ross, Don, "Game Theory", *The Stanford Encyclopedia of Philosophy* (Winter 2012 Edition), (ed. Edward N. Zalta).[Online]. Disponible: <http://plato.stanford.edu/archives/win2012/entries/game-theory/>
- [4] S. Vajda, "Mathematical Games and How to Play Them", Ellis Horwood Series in Mathematics and Its Applications, Prentice Hall PTR, Diciembre 1992.
- [5] J. Bucknall, "Behind the Minimax Algorithm", *PCPlus*, vol. 291, pp. 85-87, Febrero 2010.
- [6] P. Spronk, "Adaptive Game AI", Ph.D. thesis, Maastricht University, Maastricht, Limburg, Países Bajos, 2005.
- [7] F-h. Hsu, "Behind Deep Blue: Building the Computer that Defeated the World Chess Champion", *Princeton University Press*, 2002.
- [8] J. Schaeffer, "One Jump Ahead: Challenging Human Supremacy in Checkers", Springer-Verlag, 1997.
- [9] M. Buro, "The Othello Match of the Year: Takeshi Murakami vs. Logistello", *ICCA journal*, Vol. 20, No.3, pp. 189-193.
- [10] S. Woodcock, "Game AI: The State of the Industry", *Game Developer Magazine*, Vol. 6, No. 8., pp. 35-43, Agosto, 1999.
- [11] D. Silver, "Cooperative Pathfinding", en *AI Game Programming Wisdom 3*, Cengage Learning, 2006, pp. 99-111.
- [12] P. Spronk, F. Den Teuling, "Player Modeling in Civilization IV", en *Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, pp. 180-185, AAAI Press, Menlo Park, CA.
- [13] D. Isla, "Handling Complexity in the Halo 2 AI", en *Proceedings of the Game Developers Conference 05*, Marzo 2005. [Online]. Disponible: http://www.gamasutra.com/view/feature/2250/gdc_2005_proceeding_handling_.php
- [14] A. Champandard, "Behaviour Trees for Next-Gen Game AI", *AI Game Dev*, Diciembre 2008. [Online]. Disponible: <http://aigamedev.com/insider/presentations/behavior-trees/>
- [15] A. Champandard, "The Crisis of Integrating Next-Gen Animation and AI", *AI Game Dev*, Diciembre 2007. [Online]. Disponible: <http://aigamedev.com/open/review/crisis-animation-integration/>
- [16] M. Booth, "The AI System of Left 4 Dead", en *Proceedings of the Fifth artificial Intelligence and Interactive Digital Entertainment International Conference* (eds. Christian Darken and Michael Youngblood), Octubre 2009.
- [17] L. Lidén, "Strategic and Tactical Reasoning With Waypoints", en *AI Game Programming Wisdom*, Charles River Media, 2002, pp. 211-220.
- [18] P. Siebers, "Academic AI vs Game AI", *GTMSSIG Meeting*, Julio 2013. [Online]. Disponible: <http://www.cs.nott.ac.uk/~pos/bemoosig/slides/BEMooSSIG-Siebers2013-07-08.pdf>
- [19] L. Dicken, "Students: Game AI vs Traditional AI", *#AltDevBlog*, Noviembre 2011. [Online].

- Disponible: <http://www.altdevblogaday.com/2011/07/11/students-game-ai-vs-traditional-ai/>
- [20] M. Barton, *Dungeons & Desktops: The History of Computer Role-Playing Games*, A K Peters, Septiembre 2010.
 - [21] Encyclopedia Gamia: The Gaming Wiki, “Beat’em up” [Online]. Disponible: http://gaming.wikia.com/wiki/Beat_%27em_up
 - [22] D. Mark, “Sharing the Sandbox: Can we improve on GTA’s Playmates?”, *AI Game Dev*, Mayo 2008. [Online]. Disponible: <http://aigamedev.com/open/articles/sandbox-improve-gta/>
 - [23] Lankveld, Schreurs, P. Spronk, J. Herik, “Extraversion in Games”, en *Proceedings of 7th International Conference CG2010* (eds. H. Jaap van den Herik, Hiroyuki Lida, AskePlaat), pp. 263-275.
 - [24] P. Spronk, I. Sprinkhuizen-Kuyper, E. Postma, “Difficcutly Scaling on Game AI”, en *Proceedings of the GAME-ON 2004: 5th International Conference of Intelligent Games and Simulation* (eds. Adbennour El Rhalibi, Danny van Welden), pp. 33-37, EUROSIS, Bélgica.
 - [25] P. Spronk, M. Ponsen, I. Sprinkhuizen-Kuyper, E. Postma, “Adaptive Game AI with Dynamic Scripting”, *Machine Learning* 63, no.3, pp 217-248, 2006.
 - [26] I. Espejo, F. Fernández, M. A. López, M. Muñoz, A. M. Rodríguez Chía, A. Sánchez, C. Valero, *Estadística Descriptiva y Probabilidad*, 3ª Edición, Universidad de Cádiz, 2006.
 - [27] C. Abt, *Serious Games*, New York: Viking Press, 1970.
 - [28] M. Zyda, “From visual simulation to virtual reality to games”, *Computer*, 38, 2005, pp. 25-32.
 - [29] A. M. Turing, “Computing Machinery and Intelligence”, *Mind*, vol. 59, pp. 433-460, Octubre 1950.
 - [30] R. Leadbetter, “Tech Evolution: How Devs Pushed Consoles to their Limits”, Games Industry International, Octubre 2012. [Online]. Disponible: <http://www.gamesindustry.biz/articles/digitalfoundry-tech-evolution-how-devs-pushed-consoles-to-their-limits>
 - [31] Boxing, AtariAge, [Online] Disponible: http://atariage.com/software_page.html?SoftwareID=882
 - [32] Karateka, MobiGames, [Online] Disponible: <http://www.mobygames.com/game/karateka>
 - [33] J. Dankoff, “Game Telemetry with DNA Tracking on Assassin’s Creed”, *Gamasutra*, Marzo 2014. [Online] Disponible: http://www.gamasutra.com/blogs/JonathanDankoff/20140320/213624/Game_Telemetry_with_DNA_Tracking_on_Assassins_Creed.php
 - [34] A. Champandard, “How to have fun in an Asylum”, *AI Game Dev*, Diciembre 2012. [Online] Disponible: <http://aigamedev.com/ultimate/video/batmanasylum/>
 - [35] J. Peel, “Batman: Arkham City players finished the game without fully understanding the combat system”, *PCGamesN*, Noviembre 2013. [Online] Disponible: <http://www.pcgamesn.com/batman-arkham-city-players-finished-game-without-fully-understanding-combat-system>
 - [36] K. Hamilton, “How Assassin’s Creed III disappointed de? Let me count the Ways”, *Kotaku*, Agosto 2012. [Online] Disponible: <http://kotaku.com/5958941/how-has-assassins-creed-iii-disappointed-me-let-me-count-the-ways>
 - [37] K. Hamilton, “Assassin’s Creed III was disappointing. How does Black Flag Stack up?”,

- Kotaku, Julio 2013. [Online] Disponible: <http://kotaku.com/5958941/how-has-assassins-creed-iii-disappointed-me-let-me-count-the-ways/1460577730>
- [38] R. González Duque, *Python para todos*, Creative Commons, [Online] Disponible: <https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>
- [39] G. Dunbar, "The LUA Tutorial", Septiembre 2013. [Online]. Disponible: http://www.gamedev.net/page/resources/_/technical/game-programming/the-lua-tutorial-r2999
- [40] Getting Started with Lua, Game Dev Geek. [Online] Disponible: <http://gamedevgeek.com/tutorials/getting-started-with-lua/>
- [41] What is ScyPy?, ScyPy.Org. [Online] Disponible: <http://scipy.org/getting-started.html>
- [42] Parallel Python Overview, [Online] Disponible: <http://www.parallelpython.com/>
- [43] C. Jorgensen, M. F. Pavkovic, R. S. Rice, F. S. Schneid, C. Scott, *Fighting Techniques of the Early Modern World AD 1500 – AD 1763*, Thomas Dunne Books,





ANEXOS

Anexo	Nombre
A	Definición y Breve Historia del Combate Melee
B	Descripción de los tipos de Movimientos en Combate
C	Flujo principal del programa
D	Depuración del historial de la matriz de transición





Anexo A.

Definición y Breve Historia del Combate Melee.

Melee, proveniente del francés: *mêlée*, se utiliza para referirse al combate cercano desorganizado que involucra un grupo de guerreros. Un combate melee se lleva a cabo cuando grupos se encasillan en combate sin tomar en cuenta las tácticas de grupo o peleando como una unidad organizada, pues cada participante libra su batalla individual.

Durante la era medieval, los torneos solían involucrar un “*mêlée*” que consistía en caballeros peleando uno a uno a pié o a caballo, divididos en dos bandos o peleando todos contra todos. El objetivo era capturar caballeros contrarios para que pudieran ser liberados a cambio de un rescate, lo que se volvió un negocio muy rentable para caballeros habilidosos como William Marshal.

Hoy en día, el término Melee se ha extendido para referirse a otros tipos de combate, como las batallas navales o de vehículos blindados, que se pelean anormalmente a muy corto alcance. Hasta la Primera Guerra Mundial los regimientos de caballería de la mayoría de los ejércitos europeos fueron entrenados en dos formas de combate montado: la carga con el choque de jinetes en masa, armados con lanzas y sables, y el melee. Posteriormente fue concebido como una turbulenta masa de combates individuales después de que las líneas enfiladas de carga se habían quebrado por el shock del encuentro inicial (puede ser analizado más detalladamente en el libro *Fighting Techniques of the Early Modern World AD 1500 – AD 1763*).

En juegos ha sido adoptado y popularizado como un término en juegos de guerra, juegos de tablero, juegos de rol y videojuegos para englobar todos los combates de corto alcance e impactando al oponente en distancias generalmente menores a uno o dos metros utilizando, aunque no solamente, puños, piez, dagas, cuchillos, bayonetas u otro tipo de armas melee o hechizos especiales con la intención de causar daño. El término fue aplicado a los juegos por primera vez por H.G. Wells en 1913 en su obra llamada “*Little Wars*”, donde el autor desarrolla la “regla melee” en su juego de guerra, y posteriormente fue popularizada por el juego “*Calabozos y Dragones*”, que incluía la “fase melee” para representar el combate de personajes ajeno a los arcos y la magia.

El término sigue aplicando para la mayoría de los juegos de rol, pero también es utilizado en el contexto de los videojuegos First Person Shooter para especificar un ataque muy cercano. Esto comenzó con el juego “*Wolfenstein 3D*”, el cual incluía un cuchillo que podía ser seleccionado del inventario como un arma. Debido al riesgo que involucraba el usar un arma melee, eran típicamente las más poderosas disponibles en términos de daño. En *Duke Nukem 3D* incluía un botón que permitía al personaje patear a los enemigos mientras se seguía portando el arma de fuego. En los juegos de RTS, las unidades de infantería que no utilizan pólvora son llamadas comúnmente unidades melee.

El género Hack & Slash se caracteriza por definir su gameplay basado casi en su totalidad en las armas y el combate melee, que ha cambiado ligeramente conforme ha evolucionado con los años pero sin alejarse demasiado del origen histórico del término. Juegos de años más recientes como *Left 4 Dead 2* (Valve – 2009), incluyen una gran gama de armas melee en su Gameplay, que suelen ser muy poderosas contra los enemigos (en este caso, zombies) pero que tienen muy corto alcance y tienen

cierta desventaja ante las armas de fuego que se pueden encontrar durante el juego.



Anexo B.

Descripción de los tipos de Movimientos en Combate.

El sistema de combate para los experimentos se acota de un modo básico para realizar las pruebas de funcionalidad.

Se define la cantidad porcentual de estamina gastada en cada tipo de movimiento:

Movimiento	% Stamina
At. Fuerte	25
At. Medio	15
At. Débil	5
Df. Fuerte	20
Df. Débil	10
Bash	15
Parry	10
Evasión	45

Cada tipo de ataque se relaciona con la energía utilizada para realizarlo, en conjunto con el tiempo necesario para su ejecución. De manera más explícita, se puede identificar, por ejemplo, que el ataque fuerte requiere el uso de ambas manos en la espada, mientras que el ataque medio sólo requiere de una mano y se diferencia del ataque débil por su velocidad y la cantidad de fuerza que requiere el combatiente que lo ejecuta.

En el caso de los tipos de defensa, existen 5 definidas, tres de ellas en lo referente al uso del armamento. La defensa fuerte se define como un movimiento en el que se utiliza fuerza para detener un ataque de gran magnitud, por lo que es necesario conservar el centro de gravedad para esperar el impacto, lo que hace que se pierda totalmente la movilidad y el uso rápido del arma que se está utilizando. La defensa débil se define como un movimiento relativamente rápida utilizando algún escudo o algún tipo de arma que permita una recuperación rápida contra la defensa fuerte. El Bash (golpear o aporrear) hace referencia al movimiento ejecutado con el escudo que realiza un rechazo en cierta dirección que sigue al ataque recibido, lo que puede hacer que el atacante pierda el balance después de realizar su ataque. El Parry (parada o bloqueo) se establece como una manera de defensa parecida al Bash con el escudo, pero en este caso con el arma empuñada, lo que permite de igual manera hacer perder el equilibrio al contrincante, con la diferencia de que la probabilidad de recibir un golpe directo es mucho más alta.

Se define la Evasión como el uso de energía para esquivar cierto tipo de ataque que represente un riesgo muy alto, esto puede ser con un pequeño salto en dirección contraria a la del ataque que se está recibiendo o el arrojarse violentamente al suelo para poder evitar el daño. Dependiendo de la magnitud de la evasión será la estamina gastada y el tiempo de recuperación, pero por ser un último recurso, se busca que la Evasión reciba una penalización extra en el uso de estamina para que el jugador no abuse de este movimiento.

A continuación las tablas con las asignaciones de cada uno de los ataques.

Ataque Fuerte	T1	T2	T3	T4	T5	T6	T7	T8
Daño	0%	0%	0%	0%	0%	30%	70%	100%
Stamina de Ataque	10%	20%	30%	50%	60%	70%	80%	100%
Stamina al Bloqueo	30%	50%	60%	70%	80%	100%	110%	120%
Bash Penalty	20%	20%	20%	20%	30%	40%	50%	60%
Counter Penalty	50%	50%	40%	40%	30%	20%	10%	5%
Efectivo	NO	NO	NO	NO	NO	NO	SI	SI

Ataque Medio	T1	T2	T3	T4	T5
Daño	0%	0%	30%	70%	100%
Stamina de Ataque	20%	40%	60%	80%	100%
Stamina al Bloqueo	40%	60%	80%	100%	120%
Bash Penalty	20%	30%	40%	50%	60%
Counter Penalty	50%	40%	30%	20%	10%
Efectivo	NO	NO	NO	SI	SI

Ataque Débil	T1	T2	T3
Daño	0%	50%	100%
Stamina de Ataque	30%	60%	100%
Stamina al Bloqueo	60%	100%	130%
Bash Penalty	20%	40%	60%
Counter Penalty	50%	30%	20%
Efectivo	NO	NO	SI

Las proporciones porcentuales de la estamina están basadas en la cantidad de estamina que consume cada tipo de ataque como un 100%, proporcional a la barra de estamina con la que cuenta el jugador. Cada uno de los ataques se define como un tipo de objeto que contendrá una matriz parecida a las anteriormente citadas con todos los valores propuestos para su evaluación.

Una vez definida las matrices se crean macros que incluyan series de 3 tipos de ataque o defensa. Se definen 2 tipos de individuo: jugador, que es quien ejecutará la macro, y el NPC que es quien intentará calcular la cantidad de estamina restante en el jugador para intentar un contraataque o una medida defensiva o evasiva por cada macro ejecutada por el jugador. Ambos mecanismos serán parte del sistema de ataque del NPC final, que están siendo probados de manera modular.

Aún están por establecerse los movimientos encadenados de mayor eficacia para el jugador, por lo que hasta el momento se considerarán solamente macros de 3 movimientos que supondrán que solamente se podrán encadenar *combos* de 4 movimientos.

Anexo D.

Depuración del historial de la matriz de transición.

A continuación se describe el método de depuración de la información antigua en la base de conocimientos del sistema.

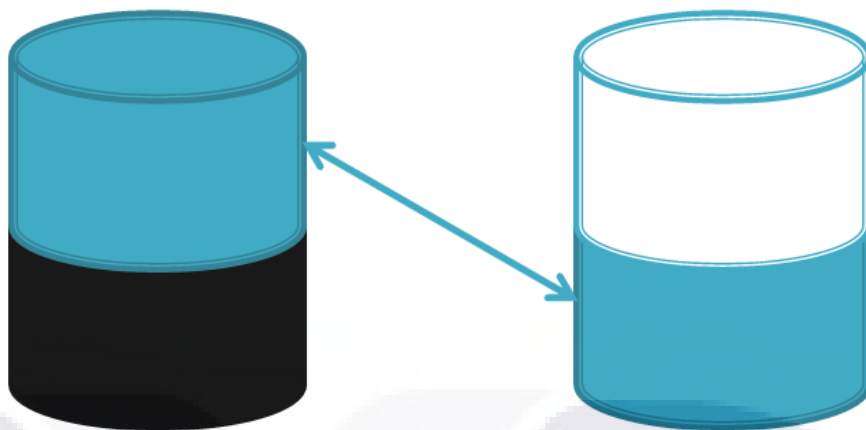
1. Sean A y B dos matrices de transición de dimensiones 4×4 con una capacidad de almacenar una cantidad t_{max} de transiciones.
2. Sea t_A y t_B la cantidad de transiciones contabilizadas para las matrices A y B respectivamente con $t_B = 0 \Leftrightarrow t_A = t_{total} \leq \left(\frac{t_{max}}{2}\right)$
3. Sean A_1, A_2, B_1, B_2 segmentos de las matrices A, B de tamaño $\left(\frac{t_{max}}{2}\right)$ donde $A_2 = B_1 \Leftrightarrow t_B > \left(\frac{t_{max}}{2}\right)$ con $B_2 = \emptyset$ y $B_2 = A_1 \Leftrightarrow t_A > \left(\frac{t_{max}}{2}\right)$ con $A_2 = \emptyset$ lo cual representa que mientras las transacciones totales en B sean menores a la mitad del máximo permitido de transiciones, el segmento 2 de la matriz A estará sincronizado con el segmento 1 de la matriz B eliminando el segmento 2 de la matriz B y viceversa.

Ejemplo:

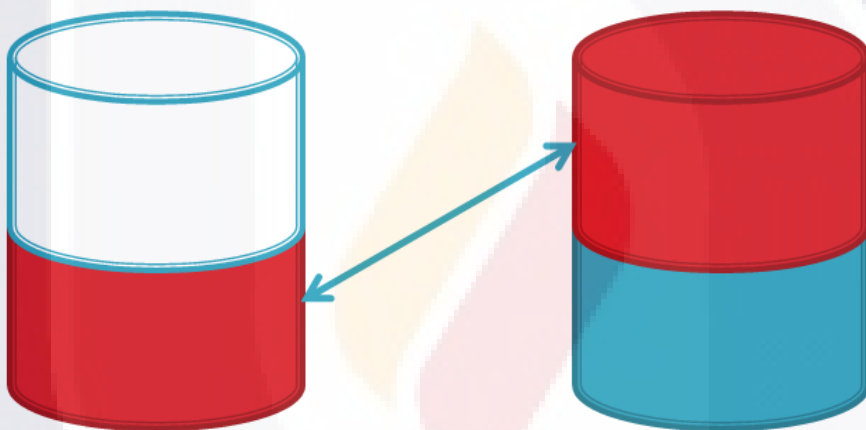
1. Se definen las matrices A y B sin información con $t_{max} = 200$ con A como matriz principal.
2. La cantidad de transiciones capturadas en B (t_B) será cero si la matriz A no ha alcanzado la mitad de transacciones máximas permitidas (t_{max}).



3. Una vez sobrepasada la capacidad del segmento A_1 , comenzará la captura de A_2 y B_1 de manera simultánea.



4. Al alcanzar la capacidad máxima del segmento B_1 , se elimina toda la información en A para comenzar la captura simultánea de B_2 y A_1



5. Se repite el paso número 3, no sin antes eliminar toda la información en B y así repitiendo el ciclo de manera indefinida hasta una condición de paro definida por el programador.

