



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“DISEÑO DE UNA ARQUITECTURA PARA INCORPORAR
EMOCIONES EN VIDEOJUEGOS”**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS
(COMPUTACIÓN)**

P R E S E N T A:

DAVID AURELIO ACEVEDO MORENO

DIRECTOR: DR. ANA LILIA LAUREANO CRUCES

México, D.F.

2009

Dedicatoria

*A mi esposa Gaby
A mi madre y a mi hermana*

Agradecimientos

Quiero agradecer a todos aquellos que de una manera u otra han colaborado en la realización de este trabajo...

A la UNAM y al Posgrado por permitirme cumplir una más de mis metas profesionales.

Al CONACYT por el apoyo económico recibido durante mis estudios.

A la Dra. Ana Lilia Laureano Cruces por su confianza y amistad.

A todos mis amigos que siempre han creído en mí y en mis capacidades.

Resumen

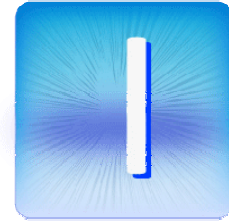
Una de las principales características de los videojuegos y ambientes virtuales es la capacidad con el que se puede modelar la realidad, lo que es aún más significativo, cuando hablamos de personajes virtuales pues cualquier persona podría evaluar el nivel de realismo al compararlo con sí mismo o con cualquier otro individuo cercano. De esta forma, identificamos la necesidad de desarrollar un modelo con un comportamiento más humano (creíble) para estos personajes virtuales.

Estudios recientes sobre la mente humana, revelan que las emociones juegan un papel importante durante el proceso de toma de decisiones, más aún, nos indican que los procesos cognitivos y las emociones trabajan juntos de manera sincronizada para formar comportamientos. Bajo esta teoría se presenta el análisis y diseño de un modelo de emociones que afecta el proceso de *toma de decisiones*. Como marco de prueba se eligen los videojuegos.

Un módulo importante en los videojuegos es aquel que controla la inteligencia artificial. Este es el módulo encargado de administrar y coordinar las acciones de los distintos agentes presentes en el juego, ayudado de múltiples técnicas como: pathfinding, maquinas de estados finitos, sistemas multiagentes, redes neuronales, lógica borrosa, etc. Este módulo proporciona la capacidad de simular comportamientos inteligentes guiando a los agentes a través de la interacción con su ambiente y transformarlo. Lo anterior con el fin de cumplir con objetivos bien definidos, como por ejemplo, eliminar al jugador.

Actualmente la *computación afectiva* pretende generar modelos y teorías para incorporar las emociones en los sistemas de software con el fin de desarrollar sistemas con mayores niveles de adaptación durante la interacción *humano-computadora*. En este trabajo, se realiza una revisión de algunas de las teorías que giran sobre diferentes formas de hacer un tratamiento informático de la valoración de las emociones. Como parte del objetivo principal, se adapta el módulo computacional del comportamiento con emociones como parte del sistema que maneja la inteligencia artificial en lo videojuegos.

Como marco de prueba de estos conceptos se desarrolla un sencillo videojuego, llamado *Dave Wars*. En este ambiente los personajes cuentan con distintas personalidades y responden a los estímulos del entorno a través de su proceso de toma de decisiones. Utilizando para ello su propia: percepción, conocimiento, creencias y estados emocionales. Debemos hacer notar que nuestro objetivo al incorporar emociones se centra en incrementar la *credibilidad en el comportamiento del agente* al hacer que este sea más humano. Para lograrlo se enriquece el proceso de toma de decisiones de los agentes, con una estructura cognitiva de emociones. Este proyecto no se enfoca en la forma final de como son expresadas estas emociones, ya que lo consideramos un área de investigación diferente.



Índice general

1. Introducción	1
1.1. Emociones y su relación con el comportamiento	1
1.2. Videojuegos, Personajes Virtuales y Emociones	3
1.3. Motivación	4
1.3.1. Realismos en Videojuegos	4
1.3.2. Interacción Humano-Computadora	5
1.3.3. Comportamiento en Agentes Autónomos	5
1.3.4. Mente Humana e Inteligencia Artificial	6
1.4. Objetivos	6
1.4.1. Enriquecer las conductas de personajes virtuales	7
1.4.2. Diseñar una arquitectura afectiva	7
1.4.3. Implementar las ideas propuestas	8
1.5. Nuestra Aproximación	8
1.6. Estructura de éste Trabajo	9
2. Computación Afectiva	11
2.1. Emociones y su rol en la toma de decisiones	11
2.1.1. ¿Qué son las emociones?	11
2.1.2. La Inteligencia Emocional	14
2.2. ¿Qué es la Computación Afectiva?	19
2.2.1. Dilemas de las computadoras con emociones	19
2.2.2. Beneficios	21
2.2.3. Clasificación	23
2.3. Teorías y modelos para simular emociones	24
2.3.1. La Teoría OCC	25
2.3.2. El modelo EMA	35
2.3.3. Otras Arquitecturas y Modelos	38
2.4. En nuestro caso de estudio	40

3. Videojuegos Afectivos.....	43
3.1. Estado del arte	43
3.2. Componentes de un videojuego.....	47
3.2.1. Servicios de IO	47
3.2.2. Motor de Render.....	48
3.2.3. Motor de física y detección de colisiones.....	49
3.2.4. Administrador Jerárquico de escena	49
3.2.5. Servidor de sonido.....	50
3.2.6. Ejecutor de guiones	50
3.2.7. Servidor de animación	50
3.2.8. Servicios de red.....	51
3.2.9. Motor de Inteligencia Artificial	51
3.3. La inteligencia artificial en los videojuegos.....	52
3.3.1. Descripción	52
3.3.2. Percepción.....	53
3.3.3. Selección de acciones	54
3.3.4. Movimiento autónomo.....	54
3.4. Técnicas utilizadas en los videojuegos	56
3.4.1. Patrones	56
3.4.2. Máquinas de Estados Finitos.....	56
3.4.3. Sistemas Basados en Mensajes.....	57
3.4.4. Sistemas de Información Basados en Posición.....	58
3.4.5. Arquitecturas basadas en metas	58
3.4.6. Otras técnicas.....	59
3.5. ¿Qué es un videojuego afectivo?	60
3.5.1. Definición	60
3.5.2. Clasificación.....	61
3.5.2.1. Asísteme	61
3.5.2.2. Rétame.....	62
3.5.2.3. Emocíname	63
3.6. Técnicas para representar emociones en videojuegos	63
3.6.1. Diseño.....	63
3.6.2. Simulación de Emociones usando Máquinas de Estados Finitos.....	64
3.6.3. Variables, objetos y los Sims	65
3.7. Nuestro caso de estudio.....	67
4. Arquitectura Emotiva.....	69
4.1. Necesidad de una Arquitectura Emotiva en Videojuegos.....	69
4.2. Descripción General	70
4.3. Un Ejemplo Concreto	72
4.3.1. El Entorno	72
4.3.2. Los Agentes.....	73
4.3.3. La Emoción Activa	74

5. Diseño de una Arquitectura Emotiva Aplicada a un Caso Específico	75
5.1. Introducción.....	75
5.2. Entorno.....	76
5.3. Metas	77
5.4. Normas.....	78
5.5. Acontecimientos	79
5.6. Emociones.....	82
5.6.1. La Emoción de estar Contento	83
5.6.2 La Emoción de Desprecio.....	84
5.7. Macroestructura de Valoración.....	85
5.8. Modelo Cognitivo.....	89
5.8.1. Comportamientos	90
5.8.2. Grafos Genéticos.....	94
5.9. El Agente y sus Emociones.....	96
6. Desarrollo de la Plataforma de Experimentación	103
6.1. Herramientas de Desarrollo.....	103
6.1.1. XNA	103
6.1.2. Arquitectura de XNA	104
6.1.3. XNA Framework	105
6.2. Arquitectura de la Implementación	107
6.2.1. Administración del flujo del juego	107
6.2.2. Jerarquía de Entidades.....	109
6.2.3. Modelo de Mensajes.....	110
6.2.4. Cola de Objetos	113
6.2.5. Modelo de Colisiones.....	113
6.2.5.1. Octree	114
6.2.5.2. Colisión mediante esferas	115
6.3. Dibujo de la Escena	115
6.3.1. Modelo de Cámara	116
6.3.1.1. Cámara Mirar hacia.....	116
6.3.1.2. Cámara libre.....	116
6.3.2. High Level Shading Language.....	117
6.3.2.1. BasicEffect	117
6.3.2.2. InstancedEffect	118
6.3.3. Preprocesamiento y posprocesamiento.....	119
6.3.3.1. Skybox	120
6.3.3.2. Posprocesamiento.....	120
6.4. Modelos Tridimensionales y Animación	121
6.4.1. Modelos Tridimensionales estáticos	121
6.4.2. Modelos de Agentes	122
6.5. Movimiento autónomo	124
6.5.1. Comportamientos dirigidos	125
6.5.1.1. Buscar	125

6.5.1.2. Huir.....	126
6.5.1.3. Llegar	126
6.5.2. Planeación de rutas	126
6.5.2.1. Mapas de navegación	126
6.5.2.2. Funcionamiento del planificador de rutas.....	127
6.6. Modelo de metas.....	129
6.6.1. Composición de las metas.....	129
6.6.2. Jerarquía de metas para los agentes	131
6.7. Modelo Emocional.....	134
6.7.1. Generador de Acontecimientos	134
6.7.2. Entidad emocional	135
6.7.2.1. La clase soldado	135
6.7.2.2. Cerebro emocional.....	137
7. Presentando: <i>Dave Wars</i>.....	139
7.1. Descripción de Pantallas	139
7.1.1. Menú Principal	139
7.1.2. Pantalla de Ayuda.....	140
7.1.3. Elementos del juego	141
7.1.4. Detalle de la interfaz del juego.....	143
7.1.5. Pantalla de término.....	145
7.2. Control de Usuario	145
7.2.1. Control de Xbox 360	145
7.2.2. Control del jugador	146
7.2.3. Control de pruebas	147
7.3. Funcionamiento del sistema.....	147
7.3.1. Aplicación	147
7.3.2. Identificación del modelo emocional	149
7.3.2.1. Un estado sin emoción.....	150
7.3.2.2. La emoción de contento	152
7.3.2.3. La emoción de desprecio	153
8. Conclusiones y Trabajo Futuro	155
Apéndice A	161
Referencias.....	163



Capítulo 1

Introducción

1.1. Emociones y su relación con el comportamiento

Cada día de nuestras vidas nos enfrentamos a diferentes tareas y problemas, algunos cotidianos y simples, otros complejos y de gran importancia; por ejemplo, la selección de los zapatos que usaremos el día de hoy, la preparación de la comida, la organización de las actividades que tendremos durante la semana o tal vez una decisión laboral de la cual dependa nuestro futuro económico. Todas estas situaciones producen una gran cantidad de actividad en nuestro cerebro, sin embargo, pocas veces nos detenemos a pensar en estos procesos y las implicaciones que cada aspecto no sólo del ambiente sino de nosotros mismos tienen en nuestras decisiones.

En el libro *Handbook of Cross-cultural Psychology: Basic Processes and Human Development* [BERJ97] se hace referencia al concepto de cognición como cada proceso por el cual los individuos obtienen y utilizan el conocimiento. Procesos por los cuales las personas comprenden su entorno y logran adaptaciones exitosas hacia él.

Teniendo en cuenta las características de cada uno de los ejemplos antes mencionados y en general cualquier problema, podemos intuir que se requiere de una contextualización. Además de información que proviene de la percepción y del almacenamiento previo. Lo anterior junto a otros factores nos ayudan a tomar decisiones que conllevan acciones. Entonces podemos afirmar que la cognición es un proceso complejo que depende de muchos factores pero que, sin duda alguna, es determinante en el comportamiento de cada individuo.

Regresando al ejemplo de la selección de un par de zapatos. Supongamos que un hombre de negocios es citado a una junta importante con nuevos inversionistas. Esa mañana mientras él se prepara para salir de su casa tiene que decidir si usar un par de zapatos cómodos que se pone todos los días para trabajar o puede escoger entre otro par de zapatos que sabe que le causan dolor en los pies pero lucen más nuevos y elegantes. Probablemente bajo este contexto esta persona calzará los zapatos más elegantes pues sabe que en la junta que tendrá debe presentar una buena apariencia. Este ejemplo sencillo nos muestra como una persona toma la información de su ambiente, en este caso la apariencia de los zapatos, recupera de su memoria algunos datos recordando que cierto par de zapatos le causan molestias en sus pies, y por otro lado, toma en consideración información extra que le dice que la apariencia es importante para dar una buena impresión.

Ahora vayamos un paso más lejos, supongamos que la junta fue todo un éxito y el futuro económico de esta persona esta asegurado, entonces podríamos afirmar que se encuentra muy feliz por lo sucedido, esa misma tarde su hijo le informa que ha reprobado una materia en la escuela, ¿Cuál sería su reacción?, antes de responder a esta pregunta debemos plantear el caso opuesto en que la junta con los inversionistas en realidad fue un fracaso y su empresa está en riesgo de caer en bancarrota, lo cual deja a este hombre con gran frustración y molestia durante el resto del día, entonces ¿cómo respondería a la noticia de su hijo?. Es claro que influyen muchos factores, por ejemplo, si no es la primera vez que su hijo reprueba en la escuela, sin embargo, podríamos hacer la suposición de que en el primer caso el padre podría percatarse de que no ha pasado suficiente tiempo con su hijo para ayudarlo en la escuela y en cuyo caso puede ofrecerle ayuda y mostrarse comprensivo, por el contrario, en la situación en donde las cosas no han salido bien para él, la noticia de su hijo puede ser un motivo más para aumentar disgusto por lo que sin mucha discusión castigará a su hijo por no haber estudiado lo suficiente.

Con facilidad podemos observar que en la situación anterior no sólo se llevan a cabo procesos cognitivos durante el proceso toma de decisiones, sino que uno de los aspectos que influyen en esos procesos son las emociones que experimenta el individuo [LOEG03]. De hecho algunas investigaciones demuestran que la cognición y las emociones no son elementos separados, sino que ambos son cruciales en la inteligencia humana [DAMA94].

Desde el punto de vista de la inteligencia artificial y dada la afirmación de que las emociones afectan nuestro comportamiento, entonces se abre un prometedor camino en donde vislumbramos a las emociones como parte esencial en las nuevas técnicas y algoritmos, ya sea para la toma de decisiones, la planeación o incluso una simulación más realista del comportamiento humano que proporcione a los sistemas de software de más información, en este caso las emociones, lo que repercutirá en una forma más *inteligente* de realizar sus tareas.

Como es de esperarse estas ideas, traen muchas implicaciones e ideas, abriendo un nuevo campo de investigación llamado *computación afectiva (CA)*. En el presente trabajo se expondrán estos conceptos, no sólo la importancia de las emociones, sino también algunas teorías que tratan de establecer un modelo informático para su tratamiento y aplicación.

1.2. Videojuegos, Personajes Virtuales y Emociones

Actualmente los videojuegos y el entretenimiento interactivo forman parte de la cultura moderna. La Asociación de Software de Entretenimiento (ESA) indica que tan solo en Estados Unidos en el año de 2006 la industria tenía un valor superior a los siete billones de dólares y estimaban que para el 2007 cerca del 41% de los norteamericanos habían comprado un videojuego o tenían planes de hacerlo [ESA07]. Por otro lado la empresa *Research And Markets* informa que la industria de los videojuegos en México esta creciendo y estima que podría alcanzar el valor de un billón de dólares para el final de la década [RAM07].

Hace un par de décadas la gente solía reunirse en sitios públicos como centros comerciales, bares o lugares especializados donde podían jugar videojuegos utilizando grandes máquinas llamadas *arcadías*. Conforme las computadoras personales fueron ganando terreno, cada vez más personas tenían la posibilidad de jugar en sus propios hogares lo que incluso llevó a la creación de sistemas de cómputo especializados llamados de una manera genérica consolas. Esta forma de entretenimiento digital crecía en popularidad cada día y se beneficiaba de muchas áreas como la creación de microprocesadores más eficientes y poderosos que en conjunción con nuevos algoritmos y técnicas no solo permitían la visualización de algunos puntos de colores, sino que ahora generaban imágenes en tres dimensiones basados en modelos físicos con lo que era posible generar mundos virtuales de gran realismo. Tanto se han involucrado los videojuegos en la vida cotidiana que prácticamente cualquier teléfono celular tiene incorporado alguno de ellos. De esta forma cualquier persona puede jugar sin importar la hora o el lugar.

A medida que los videojuegos iban evolucionando en el aspecto técnico, también lo hicieron en el aspecto creativo. En los primeros juegos solo existían reglas sencillas que se debían seguir para alcanzar un objetivo. Posteriormente se inventaron personajes, los cuales indudablemente tenían una historia detrás que les ayudaba como contexto para colocarlos en las situaciones específicas del juego. Por razones técnicas estos personajes no eran muy elaborados visualmente por lo que eran poco expresivos, sin embargo, esto cambiaría rápidamente.

El incremento en la cantidad de colores y pixeles que se podían mostrar permitió que el aspecto creativo fuera más lejos, no solo de los personajes mismos sino también de las historias que los rodeaban.



Fig. 1.1 Evolución del personaje Mario Bros.

Tomado de <http://www.geekstir.com/gaming/the-evolution-of-mario/>

Igual que en las películas con actores reales, los personajes en los videojuegos podían ahora tener historias complejas llenas de carga emocional que evidentemente eran reflejadas en el aspecto visual pero esto solo traería aún más retos para los desarrolladores pues se necesitaban de texturas cada vez más detalladas, modelos tridimensionales más complejos, actores profesionales que grabaran las voces, sistemas de captura de movimiento, etc.

En los videojuegos de última generación, el reto no sólo está en cómo poder hacer estos personajes virtuales más realistas visualmente hablando, sino también cómo podríamos hacer que su comportamiento fuera más humano, es decir que respondan a las acciones del usuario o los cambios del entorno de forma parecida como lo hace una persona del mundo real, afectada por: sus antecedentes, la situación que presenta, su personalidad y sus emociones.

1.3. Motivación

1.3.1. Realismos en Videojuegos

Dada la tendencia y la calidad actual de los videojuegos encontramos muchos intentos por mejorar cada aspecto de éstos. Tecnologías utilizadas por desarrolladores, por ejemplo el *Unreal Engine 3* [UNRT08], que permiten

incrementar el realismo en áreas como la visualización, animación, audio, sistemas de partículas y simulación física. Por otro lado, empresas como *Presagis* [PREA08] se dedican a crear herramientas específicamente para cubrir tareas de inteligencia artificial, sin embargo, creemos que aún no existe un nivel de credibilidad aceptable, un ejemplo de esto lo tenemos en juegos recientes (septiembre de 2007) y de gran impacto, como *Halo 3* [HALO07] [BUNG08], donde si el jugador dispara a sus aliados estos solo responderán con un dialogo similar a: “¡Hey! Fíjate dónde disparas”, y sus acciones no se verán afectadas en ningún sentido, lo que en una situación más creíble, podría incluso provocar una traición hacia el jugador si éste no tiene cuidado.

Alexander Nareyek [NARA07] señala que no hay nada de malo con las técnicas tradicionales de IA en los videojuegos como las expuestas en [DELM00] [BUCE05] y [SCHB04], sin embargo, es necesaria una IA de más alto nivel que utilice estas técnicas de bajo nivel, con el fin de crear agentes más inteligentes y de mayor credibilidad.

1.3.2. Interacción Humano-Computadora

Indudablemente, podemos dar por hecho que hay un importante nivel de interacción entre un jugador y la computadora que corre el programa específico de un videojuego. Esta interacción es inherente e inevitable dadas las características de los videojuegos. Si observamos la forma en cómo se realiza esta interacción y como los videojuegos involucran al jugador mediante técnicas narrativas y mediante la evocación de sus emociones [FRED03], podemos ver entonces, nuevas aproximaciones del uso de las emociones. Y éstas pueden también beneficiar a otros sistemas para mejorar su relación con el usuario.

Los Sistemas de Enseñanza Inteligentes (SEI), como se describe en [LAUA00₂], pueden ser vistos como agentes activos que adaptan sus estrategias de enseñanza basándose en los cambios que percibe provenientes del usuario, cuando este último se encuentra en un proceso de aprendizaje. Dado el dominio de los SEI, que incluyen la inteligencia artificial, la pedagogía y la psicología cognitiva, éstos pueden aprovechar la inclusión de emociones (desde el punto de vista del usuario) dentro del proceso de toma de decisiones que llevarán al alumno a un proceso de enseñanza-aprendizaje más efectivo y afectivo [LAUA08].

1.3.3. Comportamiento en Agentes Autónomos

Es claro que la computación afectiva tiene un alto impacto en los *agentes personificados (embodied agents)* que comúnmente vemos en los videojuegos y, como vimos anteriormente, también encontramos gran relevancia en aquellos agentes encargados de la interacción entre humanos como parte de las

interfaces de los sistemas. Sin embargo, es claro que las aplicaciones no se centran solo en estas áreas, los *sistemas multiagentes* [GOMJ05] se construyen con la finalidad de resolver un problema dado mediante técnicas de cómputo distribuido, usando agentes con cierto grado de inteligencia para que interactúen, cooperen y/o compitan entre sí, en un entorno que les permita resolver el problema en cuestión. Esta noción de interacción social entre agentes nos muestra una analogía al trabajo en equipo entre seres humanos. En este sentido, si esperamos que los agentes trabajen como lo hacemos nosotros, no entonces ¿deberían tener otras cualidades que les ayuden a tomar decisiones más inteligentes, como son las emociones?

Paolo Petta y Robert Trappl señalan en [PETP01] con relación a las emociones y los agentes, que es poco probable que el comportamiento autónomo inteligente de un sistema pueda ser alcanzado sin emociones, más aún, mencionan que la interpretación emocional puede ser un incentivo evolutivo hacia el desarrollo de capacidades cognitivas de mayor sofisticación.

1.3.4. Mente Humana e Inteligencia Artificial

Desde que el hombre tomó conciencia de sí mismo, hemos tratado de obtener respuestas del funcionamiento de nuestro cerebro: cómo se realizan los procesos cognitivos, de dónde surgen nuestras emociones, esclarecer la evolución de nuestra inteligencia, y finalmente cuál es la relación de todas ellas cuando pensamos en los individuos que somos. Con el fin de dar respuesta a estos aspectos se han estudiado desde distintas disciplinas, por ejemplo: la psicología y la neurociencia. Por otro lado, en el área computacional, la inteligencia artificial que tiene entre sus propósitos principales simular la inteligencia humana [RUSS04] se apoya en los descubrimientos de las anteriores con el fin emular algunas características y capacidades de nuestro propio razonamiento, dónde un punto importante es el contar con la capacidad para resolver problemas.

De esta forma podemos hacer una analogía entre los investigadores que analizan procesos químicos del cerebro para tratar de comprender su funcionamiento, y los investigadores del área de IA que realizan modelos matemáticos que permiten modelar nuestro razonamiento e inteligencia con el fin de resolver problemas en sistemas de software. Y en este sentido consideramos el estudio de las emociones y su relación con el comportamiento humano es de vital importancia para comprender la conducta humana.

1.4. Objetivos

En esta sección se hace una mención de los principales objetivos que nuestra investigación persiguió y los cuales se han desarrollado a lo largo del presente trabajo.

1.4.1. Enriquecer las conductas de personajes virtuales

Al guiarnos por la búsqueda de credibilidad, tanto en videojuegos como en ambientes virtuales, se han incorporado las emociones de distintas maneras, ya sea usando historias envolventes o con personajes cuidadosamente diseñados que buscan la empatía con el usuario a través de expresiones, el diálogo o la interacción.

La expresión corporal y facial de las emociones en personajes virtuales es un área con mucha investigación. En el caso de estudio se pretende un enfoque diferente, dónde se incorporan las emociones en un nivel más profundo dentro del comportamiento de los personajes. Este será un componente primario que permita un comportamiento emergente durante el proceso de toma de decisiones. Cambiando el orden de importancia de las metas que el agente persiga según los sucesos que se presenten en su entorno y la valoración de éstos. Por lo anterior las conductas que se produzcan serán diferentes.

Hemos elegido la Teoría propuesta por Ortony, Clore y Collins [ORTA88] por ser una de las teorías que dan claridad sobre el análisis y diseño para el desarrollo de una estructura cognitiva, que permita reflejar las emociones. Esta teoría permite analizar los sucesos del entorno a través de tres enfoques: 1) las metas y sus consecuencias, 2) los agentes y sus normas, 3) los objetos y sus atributos de atracción o repulsión.

Nuestro objetivo es enriquecer el comportamiento emergente de un agente virtual con una estructura cognitiva de emociones. Con el fin de volverlo más creíble.

1.4.2. Diseñar una arquitectura afectiva

Existen una diversidad de teorías y modelos relacionados con las emociones, cada uno con un enfoque distinto, lo cual no necesariamente puede ser aplicable a un sistema como lo es un videojuego. En el caso de los videojuegos se cuenta con características propias y restricciones bien definidas como lo son: el uso de la memoria y el tiempo de procesamiento. Lo anterior nos obliga a realizar un análisis que tome en consideración las restricciones mencionadas.

Nuestra investigación pretende diseñar una arquitectura basada en aquellas utilizadas por la industria que demuestra ser viable no solamente en el uso de videojuegos [DELM00] [BUCM05] [SCHB04]. Esta arquitectura permite introducir los conceptos que se extraen de la teoría de la emoción analizada, así como de la teoría de agentes y teorías conductuales.

En este caso nuestro objetivo se centra en el análisis y diseño de un comportamiento emergente que incluya emociones y pueda ser implementado dentro de los elementos considerados en los videojuegos. Siendo esta metodología transportable a cualquier otro tipo de sistema.

1.4.3. Implementar las ideas propuestas

Para ejemplificar y comprobar nuestra propuesta definiremos un videojuego que incorpore los elementos de diseño planteados. Este sistema mostrará las conductas de los agentes en un entorno dinámico en situaciones, donde claramente su estado emocional definirá las acciones que realizarán en su entorno.

Nuestro objetivo en este punto es obtener un sistema funcional basado en la arquitectura afectiva propuesta que incluye un grupo de agentes, interactuando dentro de un entorno virtual. El sistema incluye el diseño de un comportamiento emergente para uno de los agentes. Dicho agente es afectado por una estructura cognitiva, previamente diseñada, que influye en su comportamiento por medio de las emociones activas.

1.5. Nuestra Aproximación

Hasta ahora se ha expuesto una breve idea de cómo las emociones afectan el proceso de toma de decisiones, además identificamos la necesidad de que personajes virtuales den un paso más hacia el realismo, *no en el aspecto visual, sino a través de un comportamiento emergente que le permita incluir las emociones dentro de la valoración de los eventos del entorno.*

En el presente trabajo, haremos una revisión de algunas teorías y modelos computacionales que intentan hacer uso de las emociones como parte imprescindible en el proceso de razonamiento de un sistema inteligente. Es necesario aclarar que este proyecto está enfocado en la aplicación de un modelo computacional de emociones, no pretende de ninguna manera proponer una nueva teoría sobre la personalidad, las emociones, su origen o relación con los procesos cognitivos, esa es una tarea más adecuada para las áreas de la psicología cognitiva, la neurociencia o alguna otra área afín. Además, veremos las ventajas y desventajas de las distintas aproximaciones en la inclusión de emociones en sistemas de cómputo, así como algunas aplicaciones.

Posteriormente, analizaremos las arquitecturas y componentes que forman un videojuego, así como las técnicas actuales que se usan en los módulos de inteligencia artificial, para controlar a los personajes en un mundo virtual y dotarlos de su apariencia autónoma.

Sabemos que en un videojuego existen una gran cantidad de detalles que en conjunto actúan en beneficio del realismo como lo son los efectos físicos, la interacción con el ambiente, las voces y las gesticulaciones tanto faciales como corporales de los personajes. Sin embargo, en este trabajo no pretendemos abordar ninguno de estos aspectos, nuestro objetivo es mostrar como utilizando un modelo computacional de las emociones el personaje virtual tomará decisiones respecto a sus objetivos de una forma realista y coherente con base en su estado actual, sin que necesariamente estos se vean reflejados de alguna manera, por ejemplo, mediante expresiones faciales.

Bajo este contexto se ha desarrollado un entorno en el que interactúan agentes que poseen un comportamiento simplificado. Este sistema, envuelto en el contexto de videojuego, es utilizado como laboratorio de prueba para la capacidad de adaptación de acuerdo a los diferentes sucesos.

1.6. Estructura de éste Trabajo

En el *capítulo II*, se presenta en detalle las ideas que relacionan a las emociones y su rol en el comportamiento humano. Se hace una introducción de los conceptos relacionados con la *computación afectiva*, así como algunos modelos computacionales que incluyen a las emociones. Se hace la descripción de una arquitectura diseñada con este propósito. Al finalizar, incluiremos nuestra propuesta en la selección de los elementos para modelar las emociones que nos permitirá cumplir con nuestros objetivos.

En el *capítulo III*, se exponen los principios y técnicas más utilizadas en la industria de los videojuegos, así como el estado del arte en el que se encuentran. Se pondrá especial énfasis en el área de *inteligencia artificial* de los videojuegos para remarcar la importancia y necesidad de una nueva aproximación. En el caso de estudio, expondremos las técnicas que se utilizan en el nivel bajo como apoyo de la propuesta.

En el *capítulo IV*, se presenta una arquitectura general que pretende integrar las ideas expuestas en éste trabajo con una arquitectura probada en el área de los videojuegos.

En el *capítulo V*, se desarrolla una plataforma de experimentación, estableciendo un contexto y delimitando el dominio en el cual se aplicarán las ideas expuestas. Utilizando a los personajes determinados por el contexto, se definirá y aplicara un modelo cognitivo que definirá su comportamiento, además se identificarán las

emociones presentes, al igual que su relación con las conductas. El capítulo concluye formalizando la propuesta que incluye a las emociones como parte fundamental del comportamiento de los personajes.

En el *capítulo VI y VII*, se presenta el entorno preliminar que permite observar los primeros resultados del uso de nuestra aproximación. Se define un conjunto de situaciones y las acciones esperadas por parte de cada personaje de acuerdo al modelo cognitivo y al estado del entorno en curso.

En el *capítulo VIII*, finalmente, se exponen las conclusiones y trabajos futuros de este proyecto.



Capítulo 2

Computación Afectiva

2.1. Emociones y su rol en la toma de decisiones

Como lo ejemplificamos en el capítulo de introducción, las emociones parecen tener un alto impacto en las decisiones que tomamos cada día, en las conductas que desarrollamos en cada etapa de nuestra vida, sin embargo, antes de profundizar en el diseño de una arquitectura emocional para videojuegos, debemos esclarecer algunos conceptos relacionados con las emociones, la forma en cómo éstas: 1) afectan el comportamiento, y 2) como pueden implementarse. Consecuentemente podrá ser aplicada a los videojuegos.

2.1.1. ¿Qué son las emociones?

Para la mayoría de las personas, las emociones son algo cotidiano, una parte de nuestra vida. Pocas veces recapitamos sobre su origen o necesidad, sin embargo, cada día las experimentamos convirtiéndonos de esta forma en especialistas en su *uso*. Un pequeño niño que apenas empieza a desarrollar sus capacidades lingüísticas, es capaz de expresar enojo con un estridente llanto cuando no obtiene lo que desea, o en caso contrario alegría con una gran sonrisa por cosas simples de la vida. En el otro lado tenemos a los adultos quienes aparentemente están conscientes de su lugar en el mundo. En este último caso, experimentan una gran cantidad de emociones que incluso pueden presentarse al mismo tiempo, todo ello con aparente complejidad que da la contextualización de la situación y con una paradoja de emociones completamente opuestas ante un mismo evento.

Es obvio que como personas, somos irremediamente seres emocionales, y que, por ese motivo, nuestros comportamientos están fuertemente ligados a nuestras emociones, pero ¿qué son exactamente las emociones?

Wukmir [WUKV67] en su análisis para encontrar una definición de las emociones habla sobre la confusión que hay incluso en el área científica para explicar de forma adecuada estos conceptos. Scherer en [SCHK00] presenta un resumen de modelos históricos que incluyen a Platón, Descartes, Darwin y William James, los cuales causaron una importante influencia en los modelos actuales, sin embargo, también fueron fuente de grandes debates que a la fecha siguen causando polémica. Por otro lado, también se ha trabajado en una clasificación de las diferentes teorías modernas de las emociones, de las cuales damos una breve descripción:

- *Modelos Dimensionales*. Pretenden definir a las emociones basándose en su posición en una o varias dimensiones independientes, con lo que en su forma más simple se logran diferenciar entre emociones positivas y negativas.
- *Modelos Discretos*. Propusieron la existencia de emociones básicas y fundamentales que fueron desarrolladas por medios evolutivos.
- *Modelos Orientados al Significado*. Estos modelos pretenden descubrir la estructura de las emociones a partir del significado expuesto ya sea por su definición léxica o por patrones de comportamiento y socioculturales.
- *Modelos en Base a sus Componentes*. Asumen que las emociones surgen a partir de una evaluación cognitiva de situaciones y eventos previos donde el patrón de reacción está determinado por esta misma evaluación. Esta última es particular de cada ser humano.

Ninguna de estas teorías es única y cada una de ellas puede mostrar su validez. Sin embargo, encontramos que muchas veces parecen mezclarse al tomar en consideración ciertos aspectos comunes a ellas. Y en este punto hacemos hincapié en que en cuanto a las investigaciones referentes a los procesos mentales y la organización del cerebro humano, aún existen muchas incógnitas que deben ser resueltas. De aquí que caminemos siempre con cautela y apoyándonos en los trabajos previos y sobre todo tomando como marco de referencia alguno de estos modelos.

Las emociones de acuerdo a Reeve [REEJ07] son multidimensionales, existen como fenómenos subjetivos, biológicos, de propósito y sociales. Cuando hablamos de subjetivos nos referimos a como es que nos hacen sentir, por ejemplo, alegres o irritados. En el caso de los biológicos nos referimos a la movilización de energía para adaptarse a cualquier situación que se presente.

Por otro lado se les puede ver como agentes de propósito, ya que crean un deseo motivacional. Y finalmente tienen un aspecto social ya que comunican lo que sentimos a los demás y viceversa.

Para comprender o definir la emoción, se requiere estudiar cada uno de los cuatro dimensiones y la forma en como interactúan entre ellas. En la Tabla 2.1, se hace una síntesis de estas cuatro dimensiones.

Dimensión	Contribución a la Emoción	Manifestación
Subjetiva (Cognitiva)	Sentimientos y conciencia	Auto reporte
Biológica (Fisiológica)	Excitación, preparación física, respuestas motrices.	Circuitos cerebrales, sistema nervioso autónomo, sistema endócrino (hormonal)
Funcional (Propósito)	Motivación dirigida a la meta	Deseo de comprometerse con respuestas para enfrentar de forma apropiada la situación
Expresiva (Social)	Comunicación	Expresiones faciales, posturas corporales, vocalización.

Tabla 2.1. Dimensiones de la emoción.

Dada ésta complejidad, la emoción resulta difícil de definir. Por ésta situación y de acuerdo a Reeve, la emoción es un constructo psicológico que unifica y coordina los cuatro aspectos de la experiencia en un patrón sincronizado, dando como resultado, una reacción coherente ante una situación provocadora. Por lo que existen cientos de emociones, expresadas a través de sus combinaciones, variables, mutaciones y matices. De aquí que existan más sutilezas de las que podemos imaginar.

Por todo lo anterior entendemos que existen tantos perfiles emotivos como personas en el mundo y cada una de ellas lleva impreso, un perfil único que ahora llamaremos *mapa emocional*. Y ese, al igual que una huella digital, es único, impreso por experiencias, carga genética, aprendizaje, cultura, entre otros.

Afortunadamente, desde la perspectiva de la inteligencia artificial y dado los objetivos del presente trabajo, las diferentes teorías sobre las emociones nos ofrecen un rango de selección suficientemente amplio para buscar aquella que se adapte mejor para resolver nuestro problema. En este caso con el fin de crear una emoción sintética bajo las restricciones específicas del proyecto.

Dado que el fundamento de este trabajo se basa en las emociones es necesario contar con una definición de lo que es una emoción. En [SCHK05] el autor sugiere lo siguiente:

“Emoción es un episodio de cambios sincronizados e interrelacionados en los estados de todos o la mayoría de los subsistemas en respuesta a la evaluación de un estímulo externo o interno que es relevante a aspectos importantes del organismo.”

Aunque en ninguno de nuestros objetivos se encuentra formular una nueva teoría del origen de las emociones o su significado, encontramos útil establecer nuestra propia definición que nos ayudará a dejar claro el objeto más importante sobre el cual basamos nuestra teoría. De esta forma para el presente trabajo definimos una emoción como:

Un conjunto de variables que representan cambios interrelacionados en los estados de otros subsistemas en respuesta a la evaluación de estímulos externos e internos que son de alguna manera relevantes para el individuo.

Y es esta definición la que nos permite modelar agentes con apreciaciones propias que les permiten generar distintos estados emotivos. A partir de estos modelos se desarrollará el ejemplo que en nuestro caso toma la forma de un videojuego.

2.1.2. La Inteligencia Emocional

En este punto es importante mencionar la relevancia de las emociones para nosotros como individuos, de forma que podamos entender su relación con el comportamiento. Scherer [SCHK00] nos indica que el interés producido por las teorías psicológicas afectivas se puede agrupar en tres categorías:

1. La diferenciación y surgimiento de las emociones.
2. Patrones de respuesta por cada emoción en particular.
3. Los efectos de las emociones en otros procesos como la memoria y el juicio.

Esta última categoría es nuestro punto de partida para hablar de cómo las emociones afectan distintos procesos mentales como es el proceso *de toma de decisiones*, el cual será el foco de atención para este trabajo.

En la historia de la humanidad siempre ha existido el debate acerca de la naturaleza de las emociones en el comportamiento humano, mostrándonos predominantemente ejemplos donde las *pasiones* son una fuerza negativa en el comportamiento humano. Sin embargo, dado el reconocimiento de esta influencia han surgido incluso nuevas tendencias donde se identifica la importancia de las emociones para dominar nuestro comportamiento y mejorar nuestras vidas con el concepto denominado *inteligencia emocional* [INTE08], el

cual busca cómo una de sus metas que la gente adquiriera habilidades de autoconciencia, control emocional, motivación, empatía y socialización.

Para entender estas ideas, debemos profundizar en la naturaleza de las emociones y de nuestros propios procesos mentales de forma que podamos observar cómo ambos están profundamente relacionados.

Básicamente existen dos enfoques, el cognitivo y el biológico, y su punto de partida para la discusión es responder a la pregunta ¿cómo se originan las emociones? En el caso de la corriente biológica, sus seguidores como son: Carroll Izard, Paul Ekman, Jaak Panksepp y Robert Zajonc, mencionados en [REEJ07], argumentan que los niños responden de manera emocional a pesar de sus deficiencias cognitivas como puede ser: el vocabulario y su capacidad de memoria restringida. Cuando el niño crece, aprende lenguaje y adquiere capacidades de memoria, involucra un proceso cognitivo. Pero aún con esto se insiste en que el proceso de la emoción es automático, inconsciente y está mediado por estructuras subcorticales. Se argumenta que tienen un surgimiento muy rápido, de breve duración y pueden ocurrir de forma automática e involuntaria. Inclusive muchas veces se reacciona antes de que sepamos que fue lo que pasó.

De acuerdo a Panksepp mencionado en [REEJ07], las emociones surgen de los circuitos neuronales genéticamente heredados y que regulan la actividad cerebral.

En el caso de la corriente cognitiva, sus representantes son: Richard Lazarus, Klaus Scherer, Bernard Weiner y James Averill, mencionados por Reeve en [REEJ07], argumentan que la actividad cognitiva es un prerrequisito necesario para la presencia de la emoción. Lazarus argumenta que sin una comprensión de la relevancia personal del impacto potencial de un suceso sobre el bienestar personal, no existe razón para responder en forma emocional. Scherer argumenta que la gente procesa información sin la necesidad de experimentar una emoción. Ciertas situaciones producen emociones y otras no, y todo depende de la novedad, agrado, significado de la meta o necesidad, potencial de enfrentamiento o compatibilidad de norma. Estos cinco tipos de valoración son los que constituyen el tipo de procesamiento cognitivo que da origen a las emociones. Para Averill las emociones se comprenden mejor en un contexto social o cultural. En este último caso se muestra la emoción al producir efectos sociales, como en el caso del enojo, que permite violar una norma social, o negar la responsabilidad de una acción.

De esta forma resulta claro que es más difícil estudiar los sitios más recónditos de los circuitos cerebrales que analizar los sentimientos etiquetados de forma verbal. Sin embargo, ambos lados establecen con claridad sus respectivas posiciones. Y aquí habría que decir que ningún lado es el correcto. Son ambos los que le dan sentido al proceso de la emoción. De aquí que surjan enfoques

como el de Buck, mencionado en [REEJ07] que afirma que los seres humanos tienen dos sistemas síncronos que activan y regulan la emoción.

En este caso existe un sistema innato, espontáneo, fisiológico que reacciona de manera involuntaria a los estímulos emocionales. Y un segundo sistema cognitivo con base en la experiencia que reacciona en forma interpretativa y social.

El sistema de emoción fisiológico se presentó primero en la evolución de la humanidad (el sistema límbico), y el sistema de emoción cognitivo apareció después, conforme los individuos dependían más de su capacidad cerebral y se volvían más sociables (neocorteza).

Todo esto se refuerza con los descubrimientos realizados por LeDoux, neurólogo de la Universidad de Nueva York [GOLD97] Este investigador explica como la amígdala puede ejercer el control sobre lo que hacemos mientras la neocorteza intenta tomar una decisión. La relación entre estos dos elementos es vital para los estallidos emocionales. La investigación de LeDoux, es revolucionaria, debido a que encuentra vías nerviosas para los sentimientos que evitan la neocorteza. Entre los sentimientos que toman la ruta directa a través de la amígdala están los más primitivos y potentes, este circuito explica el poder de la emoción cuando supera a la razón.

La teoría tradicional explica que a partir de la neocorteza se envían las señales al cerebro límbico, y a partir de allí se difunde la respuesta apropiada hacia el cerebro y el resto del cuerpo. Y así como funciona la mayor parte del tiempo. Sin embargo LeDoux descubrió un conjunto de neuronas más pequeño que conduce directamente de tálamo a la amígdala, este además de aquellas que recorren la vía más larga. Esta vía pequeña y corta permite a la amígdala recibir entradas directas de los sentidos y comenzar una respuesta antes de que queden plenamente registrados por la neocorteza. La existencia de esta red permite durante una emergencia emocional, atraer y dirigir gran parte del resto del cerebro incluida la mente racional.

Herbert Simon [SIMH67] postuló en 1967 que las emociones fijan la atención a eventos importantes, viendo al sistema nervioso central como un procesador de información serial, con lo que supuso que servían como *interruptores cognitivos* que nos dicen qué es importante, de manera que nos facilita la priorización de los recursos. Instintivamente, la idea de Simon nos sugiere que hay un mediador en cada individuo que nos ayuda a tomar ciertas decisiones sobre aspectos de nuestra vida, donde incluso un análisis a conciencia podría no resolver un conflicto al enfrentar un problema determinado.

Por otro lado, Damasio, neurólogo de la Facultad de Medicina de Iowa, presenta en su libro [DAMA94] una de las investigaciones más importantes al sugerir la relación entre emociones y pensamiento basándose en una serie de casos

clínicos. Inicialmente explora el caso de Phineas Gage, el cual tras sobrevivir a una seria lesión donde una barra de hierro traspasa su cabeza, recibe un fuerte daño cerebral. Este último se refleja en una serie de cambios en su comportamiento principalmente relacionados con su carácter y habilidades sociales y, sin embargo, no muestra ningún deterioro en su coeficiente intelectual. El análisis de éste y otros casos le sugiere a Damasio una relación íntima entre las distintas áreas del cerebro:

Parece existir una colección de sistemas en el cerebro humano consistentemente dedicados a los procesos del pensamiento orientado a metas que nosotros llamamos razonamiento, y a la selección de respuestas que llamamos toma de decisión, con un especial énfasis en los dominios de lo personal y lo social. Esta misma colección de sistemas está también envuelta en las emociones y los sentimientos, y está parcialmente dedicada al procesamiento de señales del cuerpo

Para explicar sus ideas, Damasio plantea lo que él llama la *Hipótesis de marcadores somáticos* en ella argumenta que la toma de decisiones usualmente involucra una evaluación de procesos cognitivos y emocionales incentivados por la importancia de las acciones disponibles en cada situación en particular. Sin embargo, a menudo, la situación requiere de decisiones muy complejas con un alto grado de incertidumbre y ambigüedad donde incluso existen alternativas que se contraponen entre sí. En estas situaciones, los procesos cognitivos pueden no ser suficientes para proporcionar una respuesta satisfactoria, en cuyo caso, distintos estímulos que incluyen los estados afectivos sirven como un medio de motivación o evaluación adicional que permiten finalmente tomar una decisión. En los casos, expuestos en sus trabajo, de individuos con daño cerebral supone que estos estímulos adicionales que incluyen a las emociones, ya no pueden ser ligados al proceso cognitivo y por tanto afectan de manera importante la toma de decisiones.

Dentro del libro *Handbook of Affective Sciences* en el capítulo titulado *The role of affect in decision making* [LOEG03] se presenta un análisis sobre el rol de las emociones en la toma de decisiones. Los autores mencionan que existen dos tipos de influencias afectivas: las emociones esperadas y las emociones inmediatas o inminentes. Las *emociones esperadas* se basan en modelos que asumen que la gente intenta predecir las consecuencias emocionales asociadas con acciones alternativas, entonces seleccionan las acciones que maximizan las emociones positivas y minimizan las negativas. De esta forma, al experimentar una emoción negativa o creer que podría experimentarse influye en la toma de decisión, por ejemplo con la frase, *No invertiré porque no quiero arrepentirme*. Por otro lado, las *emociones inmediatas* son también una fuente de influencia en el proceso de toma de decisiones y su influencia se refleja a través de dos caminos: 1) las influencias anticipadas provocadas a partir de las emociones esperadas, y 2) Influencias incidentales (que nada tienen que ver con la

situación sobre la cuál se toman las decisiones. Estas influencias pueden provenir del ambiente o de una disposición a la emoción.

Algunas teorías dicen que el proceso de *toma de decisiones* no sólo está orientado a los resultados de la elección, sino también a la parte que implica esa elección en la autoevaluación, como puede ser considerarse un individuo competente o inteligente.

Por otro lado, aunque las emociones no afectan en todos los tipos de decisiones, solo aquellas situaciones en donde el propio estado emotivo es relevante, se sabe que su influencia depende de forma crítica de la intensidad de la emoción experimentada. Cuando las emociones se intensifican, incrementan su influencia sobre el comportamiento, de hecho, a altos niveles las emociones pueden abrumar el proceso cognitivo. Esto se explica más claramente cuando se habla del término *fobias*, un estado que produce que el individuo que la experimenta actúe de una forma irracional aunque sepa de forma consciente que no tiene motivo alguno, porque no hay un peligro real. Como lo expone Rolls [ROLE99]:

Las emociones frecuentemente parecen muy intensas en los humanos, de hecho algunas veces tan intensas que producen comportamientos que no parecen ser adaptativos, como desmayarse en lugar de escapar, quedarse congelado en lugar de evadir, o dudar interminablemente acerca de situaciones emocionales y decisiones, o quedar enamorado aún cuando puede predecirse que no hay esperanza alguna o que puede traer la ruina. El misterio no es solo que la emoción es tan intensa, sino que incluso con nuestras capacidades de razonamiento, los humanos aún se encuentran en estas situaciones, y pueden encontrar gran dificultad en producir un comportamiento efectivo y razonable para resolver la situación.

Algunos estudios sugieren que la valencia emocional, es decir, cuando consideramos una naturaleza dual que presenta un lado positivo y negativo, determina la forma del subsecuente procesamiento de información. Cuando experimentamos emociones negativas, especialmente si son intensas, nuestra atención se ve disminuida, cuando estas son positivas nuestra atención aumenta. Sin embargo, también un sentimiento positivo puede segar por lo que podríamos concluir que los extremos de experimentar una emoción bien puede afectar la percepción del individuo. En este sentido, como se menciona en [CHOE02] las personas con un alto nivel de excitación afectan su comportamiento y tienden a caer en habilidades bien aprendidas o hábitos. Incluso cuando pueden tener a su disposición información relevante, lo que para el comportamiento del agente se convierte en un comportamiento reactivo.

En nuestra sociedad abunda la comida alta en grasa, la pornografía, las drogas, etc., y aunque tenemos una amplia conciencia de la gran cantidad de cosas que son dañinas a nuestro cuerpo, la fuerza ejercida por la emociones puede influenciar a los niveles más altos de nuestros procesos cognitivos y pueden

distorsionar aquellas evaluaciones directamente relacionadas con el bienestar. Lo anterior podría sugerirnos también que el estado de humor, es decir, un estado emocional más prolongado y de menor intensidad [WIL199], así como la personalidad tienen también una influencia sobre el comportamiento, sin embargo, este análisis sale de los alcances de este proyecto, aunque se considera para un trabajo futuro.

2.2. ¿Qué es la Computación Afectiva?

Hasta aquí hemos introducido las emociones y las evidencias de su influencia en el comportamiento humano, lo que claramente nos deja ver que es un área muy extensa de investigación y con grandes incógnitas por descubrir acerca de la mente humana. Por otro lado, el área de las ciencias computacionales con énfasis en inteligencia artificial, ha buscado inspiración para sus teorías en fenómenos biológicos y psicológicos entre otros. Sabemos que una de las metas de la inteligencia artificial ha sido tratar de simular ciertos aspectos del comportamiento humano, buscando soluciones a problemas que los humanos parecen resolver con aparente e increíble facilidad, lo que incluye sus habilidades sociales y de interacción con otros individuos. Claramente, si identificamos a las emociones como un punto clave para el proceso cognitivo debemos suponer que una aproximación informática en este campo debe tomar en consideración el uso de las emociones como parte del proceso.

Aunque desde hace muchos años ya se ha planteado la idea de relacionar la informática con las emociones [MUEE85] fue Rosalind Picard [PICR97] quien, en 1995, introdujo el término *computación afectiva* que ella misma define como: *el cómputo que relaciona, surge o influye en las emociones*. De esta forma, se inicia una corriente relacionada con la noción de que las computadoras deben incluir a las emociones como parte de un sistema que pretenda incrementar sus capacidades cognitivas y de interacción humano-máquina.

2.2.1. Dilemas de las computadoras con emociones

Evidentemente nuestro pensamiento inicial acerca de las emociones, está generalmente relacionado a sus aspectos negativos. Desde pequeños aprendemos la consecuencia de los gritos de una madre encolerizada por la enorme travesura que no pudo ser encubierta, y si una emoción como esa pueda ser recreada por un sistema computarizado, entonces pocos de nosotros aceptaríamos la idea de tener una computadora que empiece a gritarnos por olvidar respaldar nuestra información y en castigo terminar las aplicaciones con que estemos trabajando. Claro, en el menor de los males tal vez solo saldríamos regañados o recibiríamos la indiferencia por parte del sistema operativo, sin embargo, vivimos en una sociedad donde las computadoras y los sistemas digitales cada vez están más presentes, controlando de manera significativa

nuestro entorno y cada aspecto de nuestra vida, ya sea en los teléfonos celulares, en nuestros sistemas de entretenimiento, las computadoras personales, el transporte, plantas de procesamiento, hospitales, etc. Queda claro que nadie querría hacer enojar a una planta de energía nuclear y dado caso ni siquiera a nuestro propio teléfono.

Estas ideas, aparentemente paranoicas, han sido explotadas incesantemente por las historias de ciencia ficción, dándonos una idea de lo peligroso que podría ser que una computadora pudiera experimentar emociones. Uno de los ejemplos más conocidos fue popularizado en 1968 por la película clásica *2001: A Space Odyssey* donde el personaje principal es la computadora *HAL 9000*, una entidad artificial encargada de la nave *Discovery*, provista de capacidades mentales similares a las de cualquier humano. En esta cinta encontramos diálogos que nos dan evidencia de la personalidad emotiva de Hal:

Reportero: Uno tiene la sensación de que Hal es capaz de tener respuestas emocionales. Cuando le pregunte acerca de sus habilidades, percibí un cierto orgullo...

Tripulación: Bueno, ella actúa como si tuviera emociones genuinas. Por supuesto, está programado de tal manera que haga más fácil para nosotros hablar con ella. Pero si ella tiene o no emociones reales es algo que no creo que alguien pueda responder.

Conforme la trama se desarrolla uno puede observar que Hal tiene la habilidad de expresar sus emociones y de percibir las emociones de otros, sin embargo, de una manera escalofriante también es dominado por lo que podríamos llamar *miedo* ante la posibilidad de ser desconectada lo que la lleva a realizar acciones que definitivamente la tripulación de la nave jamás espero. Curiosamente, la propia ciencia ficción ha tratado también de dar solución a este tipo de dilemas, donde el ejemplo más claro pueda verse en las tres leyes de la robótica propuestas por Isaac Asimov, de esta forma, si dotamos a las maquinas con emociones humanas, entonces también deberíamos poner un seguro que evite que estas puedan realizar alguna acción negativa para nosotros.

Tal vez esta aproximación sea un tanto dramatizada a la realidad, en dado caso podríamos considerar una situación más factible. Supongamos que nuestra computadora personal no experimenta ninguna emoción, sin embargo, si es capaz de reconocer las emociones del usuario, de esta forma podría en una situación hipotética identificar un estado de enfado que le permite a un programa iniciar una música suave y cambiar los colores de la pantalla a otros de manera que fomenten un estado de relajación que busque contrarrestar aquella emoción detectada permitiéndonos continuar nuestro trabajo durante más tiempo y concentración. Aunque esta podría bien ser una aplicación real, no es exenta de los problemas de la informática actual, es decir, si nuestro sistema es comprometido alguien podría obtener información sobre nuestro estado

emocional con lo que podrían ver la oportunidad de bombardearnos con publicidad relativa a mejorar nuestro estado de ánimo, o peor aún aprovechar el momento para realizar algún ataque de ingeniería social obteniendo datos útiles sobre nuestros hábitos o incluso información confidencial.

Al descubrir todos estos inconvenientes, ¿no deberían las emociones, ser evitadas? Tal vez sea más adecuado pensar en una aproximación diferente que nos permita obtener algún beneficio de un estado emocional artificial.

2.2.2. Beneficios

Hasta ahora hemos visto los problemas que la idea de que cualquier sistema computacional con emociones podría tener, sin embargo, antes de desechar la idea valdría la pena analizar también los beneficios que este concepto puede traer. Al igual que muchas investigaciones relativas al funcionamiento del cerebro humano, ésta aproximación puede ayudarnos a entender de una manera más clara los procesos relacionados con nuestros pensamientos, nuestras emociones y cómo éstos interactúan entre sí para dar como resultado los comportamientos que vivimos día con día.

Hasta aquí se han presentado distintas ideas, teorías y evidencia acerca de cómo las emociones afectan distintos procesos, en especial al proceso de toma de decisiones. De aquí que si una de las pretensiones es una computadora sea realmente inteligente, y actúe de forma similar a una persona, entonces debemos proveerle de un mecanismo de emociones parecido al del humano y lograr que interactúe con su modulo deliberativo.

El reconocimiento así como la expresión del afecto, son considerados como una característica necesaria para la simpatía y la comunicación del entendimiento, lo que es identificado como una de las grandes necesidades psicológicas del hombre. Incluso si consideramos un cachorro, éste podrá decirnos a través de ciertas pistas físicas o conductuales, que estamos enojados con él, lo que claramente no requiere de ninguna comunicación verbal, sin embargo, no tenemos esta misma relación con una computadora. *Evidentemente la interacción humano-máquina puede ser mejorada grandemente.*

Si hablamos entonces de dotar a las computadoras de la habilidad de reconocer las emociones de las personas, podríamos pensar en un sinnúmero de aplicaciones. Una empresa podría estar interesada en percibir el estado de humor de sus empleados, con información como esa podría detectar una variable importante relacionada con la productividad y en consecuencia tomar acciones pertinentes, incluso encontrar aquellos empleados con sentimientos de rencor que en un porcentaje alto resultan ser la causa de la fuga de información vital de la empresa. En otro aspecto de nuestras vidas, cuántas personas no desearían poder detectar los síntomas de una depresión, incluso antes de que ésta se

presente, un sistema podría ayudarnos a monitorizar a personas propensas a estos estados: personas de la tercera edad o que sufren de aislamiento prolongado. Lo que nuevamente podría traducirse en una acción preventiva o correctiva de parte de algún especialista o incluso nuestro sistema puede también tener la capacidad de inducir ciertas emociones, ya sea con cambios sutiles en el entorno, como la reproducción de música o la disminución de la temperatura ambiente o incluso un sistema muy complejo podría iniciar una conversación con el individuo para realizar una evaluación más detallada o iniciar una terapia a forma de psicólogo virtual. Sin embargo, como cualquier otro sistema, este puede ser vulnerable a fallas, el problema de la detección de la emoción es un proceso tan complejo que para nosotros mismos es normal caer en una suposición errónea sobre la emoción que pueda estar experimentando un individuo. Si por ejemplo, diseñamos un sistema para el reconocimiento de la emoción para una empresa que desea medir el estado de ánimo de sus empleados al iniciar su jornada laboral, consideramos entonces un conjunto de variables que nos permitan lograr esto e incluimos el ritmo cardíaco como una de ellas. Si nuestra medición, detecta un ritmo cardíaco acelerado podríamos obtener una conclusión de que el individuo está ansioso o temeroso por alguna razón, sin embargo, su estado actual pudo ser alterado por el simple motivo de que subió las escaleras demasiado rápido para llegar a su oficina. Como otro ejemplo, se han realizado investigaciones sobre sistemas de enseñanza [LAUA00], [LAUA08], [JAQP04] que tienen como objetivo identificar las emociones del alumno como la frustración, de manera que el sistema pueda tomar una nueva estrategia para evitar la deserción y mejorar el aprendizaje del tema en cuestión. Evidentemente no es un problema fácil de resolver, pero si de enorme beneficio si se utiliza adecuadamente.

Dejando a un lado el análisis y la inducción, encontramos otra aplicación al hablar de la *síntesis de las emociones*. Como individuos lo que revelamos a otros, voluntariamente o no, es en gran medida nuestra propia expresión emocional. La expresión a través de nuestro sistema motriz es usualmente involuntaria, y es una pista de nuestro propio estado, así con el tono de voz, la expresión facial, o la postura, es una de tantas formas en como mostramos a otros nuestras emociones. Entonces si conforme se han desarrollado tecnologías para crear robots humanoides o agentes conversacionales sumergidos en un ambiente virtual tridimensional, es natural que permitamos que un sistema pueda sintetizar de alguna manera un rango de emociones, que en busca de incrementar la semejanza de un robot con las personas le demos la oportunidad no solo de caminar, sino también de caminar alegremente o caminar con tristeza. Si uno de los objetivos es el desarrollo de agentes virtuales con comportamientos más realistas, entonces un módulo de emociones sintéticas le permitiría no sólo comportarse de acuerdo a sus emociones sino además expresarlas de tal manera que muestre esas señales que naturalmente son expuestas por las personas. De otra forma si estas expresiones no son coherentes con su comportamiento, entonces ya sea robot o agente virtual perderá su credibilidad, una parte esencial que le permita comunicarse de una

mejor forma con los humanos, ya que estos últimos los consideraran falsos e irreales.

2.2.3. Clasificación

Debido a las diferentes áreas donde la computación afectiva puede llegar a tener influencia es necesario establecer una clasificación que nos permita diferenciar los distintos sistemas y sus objetivos. De esta forma, Picard [PICR97] nos proporciona la siguiente clasificación de las computadoras y su capacidad relacionada con las emociones:

Computadoras	No expresan afecto	Expresan afecto
No Perciben afecto	I	II
Perciben afecto	III	IV

- I. La mayoría de las computadoras y sistemas computacionales caen en esta categoría, teniendo una capacidad nula de expresar y reconocer afecto.
- II. Los sistemas utilizan técnicas como voces sintetizadas con entonación natural, caras virtuales o agentes que permiten de alguna manera expresar su afecto.
- III. Esta categoría nos dice que las computadoras tienen la habilidad de percibir el estado afectivo de las personas. Véase los ejemplos de la sección 2.2.2. de éste trabajo.
- IV. Esta categoría maximiza la comunicación entre los humanos y las computadoras, potencialmente proveyendo verdaderamente computación “personal” y “amigable”. Esto no implica que los sistemas serán manejados por sus emociones.

En su clasificación, Picard deliberadamente omite la categoría donde las computadoras inducen emociones, pues argumenta que es evidente que éstas y otros medios ya influyen en nuestras emociones. Además la autora también se toma la libertad de excluir la categoría de las computadoras que pueden actuar con base en sus emociones por los problemas éticos y filosóficos que también ya han sido discutidos en este trabajo.

Por otro lado, Jaques [JAQP04] nos comenta que la computación afectiva se encuentra dividida en dos grandes áreas:

- I. La afectividad en la interacción entre el humano y la computadora.

II. Simulación o síntesis de emociones en las máquinas.

En la primera de estas áreas los investigadores están interesados principalmente en identificar el estado afectivo del usuario con el objetivo de generar un sistema adaptativo que pueda responder al usuario de alguna manera incluso afectivamente. Los sistemas pertenecientes a esta área tienen la capacidad de reconocimiento y expresión de emociones. En la segunda área los sistemas están interesados en la simulación de las emociones en las propias máquinas tratando de descubrir más acerca de las emociones de los humanos. Otros trabajos tratan de construir robots [MALM04] y agentes animados más realistas [PETP97] [BECC06] implementando emociones. Otros más buscan mejorar las interacciones de agentes sociales, sus habilidades de negociación y cooperación con otros agentes considerando por su puesto un estado afectivo. En estos casos los sistemas usualmente tienen la capacidad de poseer emociones y cuando es posible desarrollar incluso una nueva emoción.

Es importante para nuestro caso poder clasificar el proyecto desarrollado, de esta forma podremos delimitar los aspectos y subsistemas donde se encuentran involucradas las emociones. Recordando el principal objetivo de este trabajo, desarrollaremos una arquitectura que nos permitirá contar con agentes autónomos que serán influenciados por sus emociones para generar distintos comportamientos. Según la clasificación de Jaques nuestro objetivo plantea la idea de sintetizar un conjunto de emociones con lo que buscaremos que nuestros agentes tengan comportamientos que los doten de credibilidad y aumente la empatía con el usuario. Un resultado esperado del diseño de esta arquitectura permitirá que alguna implementación posterior tome la información producida y pueda utilizarla como una herramienta más a un conjunto de técnicas que ayuden a los agentes a expresar sus emociones. Para un agente con una representación visual humanoide, es decir, que asemeje algunas características físicas humanas, esto implicaría que puede expresar sus emociones por medio de expresiones faciales, movimientos corporales o alguna otra manera. De esta manera, tanto Jaques como Picard nos dan una categoría en la que nuestro sistema pueda expresar en cierta medida una emoción que le ayude a mejorar la comunicación con el usuario.

2.3. Teorías y modelos para simular emociones

Aunque las teorías sobre emociones no son algo nuevo, han sentado las bases para el surgimiento de nuevas propuestas y con el auge de la computación afectiva, no solo buscan explicar su origen o estructura, sino también se han propuesto diferentes modelos informáticos con la finalidad de dotar a las computadoras de emociones y de sus beneficios. A partir de estos modelos se han diseñado distintas arquitecturas concentradas en usar un modelo afectivo que les permita mejorar sus procesos deliberativos. Los sistemas multiagentes

han sido los principales beneficiados de estos esfuerzos debido no solo a los beneficios que los agentes pueden obtener en la toma de decisiones, sino además por su inherente naturaleza social.

Iniciamos esta sección con la descripción de la teoría OCC [ORTA88] que ha sido una de las bases sobre las cuales muchos modelos basan su definición de emociones. Proseguimos describiendo algunos modelos que plantean usar estos conceptos para establecer arquitecturas funcionales donde algunas indican sus virtudes en implementaciones que muestran el concepto del efecto de las emociones sobre el comportamiento de los agentes.

2.3.1. La Teoría OCC

Aunque existen muchas ideas sobre el surgimiento y función de las emociones, existe una aproximación que trata de explicar su origen a través de los procesos cognitivos. La idea es que las emociones dependen del significado percibido que se les da a las situaciones y a una valoración subjetiva que el individuo asigna basado en una serie de criterios propios [PICR97]

Una de estas ideas fue presentada por Ortony, Clore y Collins en 1988 [ORTA88] y dada su importancia no solo porque muchos otros modelos se basan en sus ideas, sino porque además es la base para nuestro proyecto, dedicaremos esta sección a explicar sus fundamentos. Los autores construyeron una teoría cognitiva de las emociones que explica su origen describiendo los procesos cognitivos que las evocan. La teoría es llamada OCC [ORTA88] debido a las iniciales de los autores.

Tomando el origen cognitivo de las emociones, éstas son resultado de un proceso de evaluación llamado *valoración*¹. La *teoría OCC* asume que las emociones pueden surgir de la valoración de tres aspectos de nuestro entorno:

- *Acontecimientos*. Son la forma en cómo percibimos que las cosas pasan a nuestro alrededor. Un cambio de estado de algún elemento en el entorno o incluso de nosotros mismos que es de alguna manera relevante.
- *Agentes*. Los agentes pueden ser personas, animales, objetos inanimados, abstracciones, instituciones, en general, cualquier cosa a la que podría objetársele alguna acción, aunque no necesariamente la haya ejecutado o incluso sea capaz de hacerlo.

¹ El proceso de valoración es un concepto adoptado por los autores de la teoría OCC que consiste en la interpretación de distintas situaciones, personas u objetos a través de distintos procesos mentales obteniendo como resultado un indicador que puede adoptar un valor positivo o negativo.

- *Objetos*. Todo aquello que es visto simplemente como objeto.

Según esta teoría existen además tres estructuras que permiten realizar el proceso de valoración, proporcionando diversos parámetros útiles para diferenciar todo aquello que ya fue percibido e identificándolo de alguna manera como positivo o negativo. Estas estructuras son:

- *Metas*. Representan el estado de las cosas que deseamos sucedan.
- *Normas*. Representan las creencias desde el punto de vista de las cuales se hacen las evaluaciones decisorias morales y otras clases.
- *Actitudes*. Constituyen la base para la valoración de la capacidad de atracción de los objetos.

De esta forma, tenemos que los *acontecimientos* son evaluados con base en su *deseabilidad*, es decir, si promueven o imposibilitan el alcance de una meta, como ejemplo, consideremos un individuo que es despedido de su actual empleo, este sería un suceso claramente indeseable para él, si consideramos que una de sus metas era ahorrar dinero para comprar un automóvil nuevo, sin embargo, si ésta persona estaba cansado de su empleo y estaba buscando una oportunidad para irse, entonces el acontecimiento parecería deseable para él.

Las *normas* son utilizadas para evaluar las acciones de los *agentes*, haciendo de la *plausibilidad* (o *censurabilidad* en caso contrario) la base primaria de su valoración. En este caso, supongamos que hemos sido testigos de cómo uno de nuestros vecinos agrede físicamente a su hijo de tres años. Generalmente pensamos en una norma que nos dice que los padres no deberían golpear a sus hijos, lo que nos permite valorar la acción del agente, en este caso nuestro vecino, como una acción censurable. Si en el mismo caso pensamos que otra persona se ha percatado de la situación y lo acusa ante la autoridad correspondiente, una norma diferente referente a que las personas deben proteger a los indefensos puede hacernos valorar su acción como plausible, es decir, que la podríamos considerar como adecuada, lo que podría causarnos una emoción de aprecio o gratitud.

Finalmente, los *objetos* son evaluados con base en el *agrado* (*desagrado* en caso contrario) que estos mismos o sus características causan en función de su *capacidad de atraer*. Podríamos pensar que un objeto como un coche antiguo podría causarnos un sentimiento de agrado por que lo consideramos como un “clásico”, aunque por otro lado una de sus cualidades, como lo es el alto consumo de combustible puede causarnos una emoción de desagrado.

La categorización de las emociones según OCC [ORTA88] se muestra en la Figura 2.1., donde se observan los tres grupos principales de emociones

relacionados con los aspectos del entorno que pueden valorarse, es decir, los acontecimientos, las acciones de los agentes y los objetos.

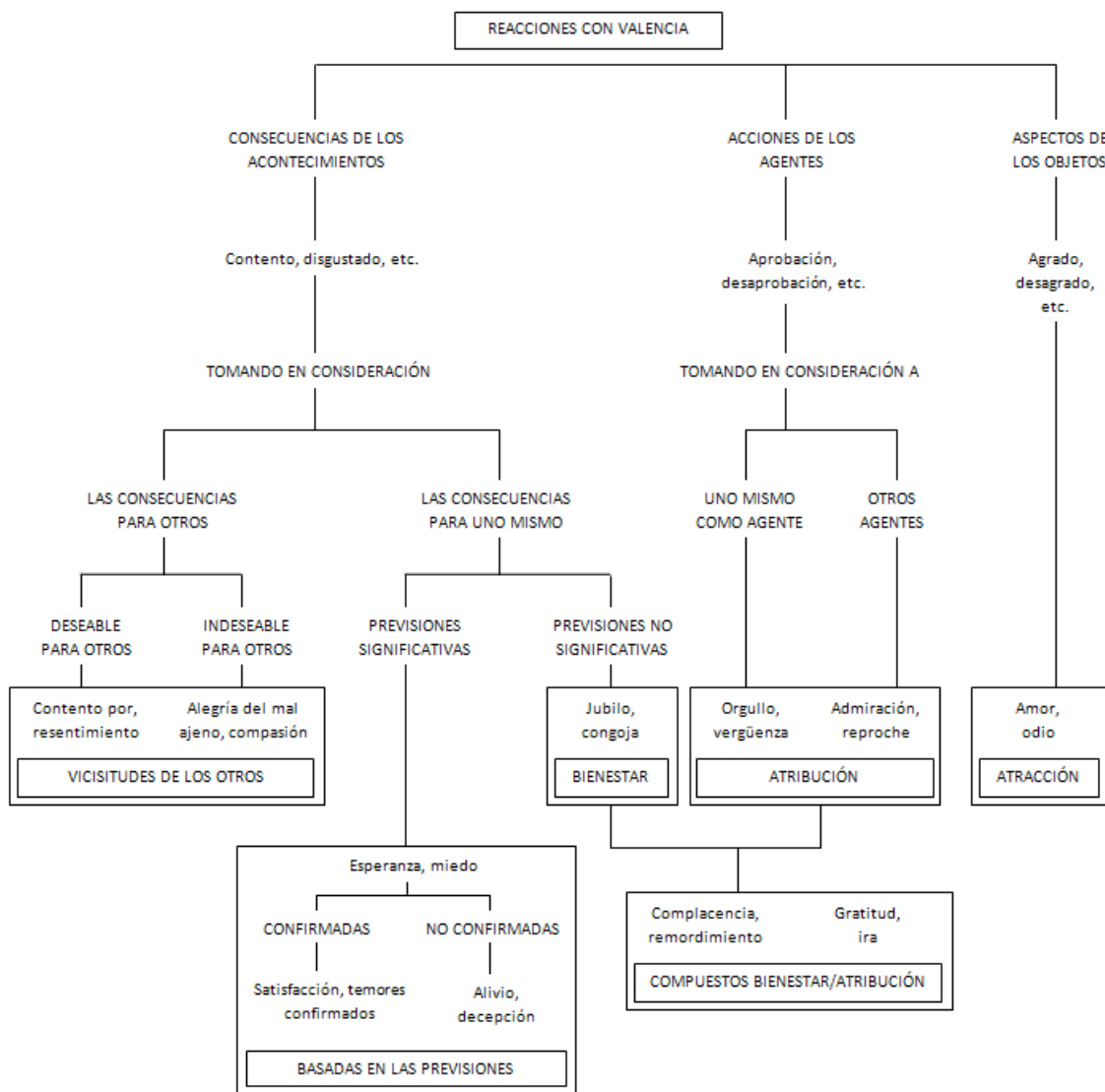


Fig. 2.1 Estructura global de los tipos de emoción

Los autores explican que cada grupo de emociones contienen un conjunto de condiciones desencadenantes que junto con la valoración producen un grado de intensidad con lo cual se puede identificar si una emoción se encuentra activa, en caso de que dicha intensidad supere un umbral dado. Además debe identificarse que existen diversos factores que afectan la intensidad de las emociones, por un lado las *variables globales*, es decir, aquellos que afectan de alguna manera a todos los grupos de emociones, y las *variables locales* que afectan a grupos de emociones o incluso a un sólo un tipo de emoción.

Algunas de las *variables globales* se describen a continuación:

- *Sentido de realidad*. Indica lo real que uno piensa que es la situación que está desencadenando la emoción.
- *Proximidad*. Depende de lo cerca que uno se sienta de la situación en el espacio psicológico. Generalmente asociado con proximidad espacial o temporal.
- *Cualidad de inesperado*. Indica que tan sorprendido está uno de la situación.
- *Excitación*. Depende del nivel de excitación previo a la situación en cuestión.

Refiriéndonos a las *variables locales*, ya hemos mencionado a tres de ellas, las cuales se refieren a los grupos principales expuestas en esta teoría, es decir, la *deseabilidad* para el grupo de *emociones basadas en acontecimientos*, la *plausibilidad* para las *emociones de atribución de responsabilidad* (o simplemente *emociones de atribución*), es decir, referentes a los agentes y sus acciones, y la *capacidad de atraer* para las *emociones de atracción*. Otras variables ligadas a grupos particulares de emociones son:

- *Probabilidad*. Refleja el grado de creencia en que un acontecimiento previsto puede ocurrir.
- *Esfuerzo*. Refleja el grado en que se han gastado recursos en obtener o evitar un acontecimiento.
- *Realización*. El grado en que el acontecimiento ocurre realmente. Ocurrencia total o parcial.
- *Afecto*. Refleja lo atraído que uno se siente por una persona.
- *Merecimiento*. Muestra el grado que uno piensa que otra persona merece lo que está sucediendo.
- *Fuerza de la unidad cognoscitiva*. Refleja cómo uno se identifica con alguna otra persona, grupo o institución que es el agente relacionado con el acontecimiento.
- *Desviación de las expectativas*. Refleja cómo la acción del agente se desvía de las normas esperadas.
- *Familiaridad*. Muestra que tan comfortable se siente una persona con la situación o con el objeto.

Aunque éstas no son todas las variables que pueden intervenir en la valoración, nos damos cuenta lo intrincado de este proceso. Y cómo aún, el acontecimiento aparentemente más simple puede ser valorado no solo desde diferentes perspectivas, sino además utilizando un conjunto diferente de variables cada vez, lo cual hace de esto un gran problema a resolver.

Uno de los principales problemas que han existido en cualquier teoría de emociones, además de su definición y clasificación, ha sido el tratar de identificar a cada emoción no solo con su nombre sino también por sus características. Para explicar este punto basta con tomar casi cualquier expresión o palabra referente a una emoción, por ejemplo, la alegría donde podemos identificar un conjunto de palabras relacionadas como lo son: contento, felicidad, gozo, júbilo, etc. El problema radica en que para dos personas aunque estas palabras podrían identificar la misma emoción con diferente intensidad, es también posible que las consideren como emociones completamente diferentes. Incluso las diferencias socio-culturales agravan esta situación aún más.

Para lidiar con este problema los autores de la *teoría OCC* crearon lo que ellos llamaron *especificación de la emoción* que es una descripción más formal y completa de cada una de las emociones propuestas. Esta caracterización se compone de las siguientes partes:

- *Identificación del tipo*. Sirve meramente como un rótulo de identificación.
- *Especificación del tipo*. Forma una descripción aproximada de las condiciones necesarias para experimentar la emoción.
- *Expresiones*. Una lista de palabras o expresiones que identifican a este tipo de emociones. Obviamente esta no es una lista completa pero puede ayudar a identificar el grupo.
- *Variables locales que afectan la intensidad*. Identificando las principales variables que tienen algún efecto sobre este grupo de emociones.
- *Ejemplo*. Finalmente se presenta un ejemplo prototípico.

Con una especificación formal disponible es posible mostrar una descripción de cada una de las emociones propuestas por esta teoría, iniciando con aquellas agrupadas en la categoría de *emociones basadas en acontecimientos*.

Emociones de júbilo

Especificación de tipo: (contento por) un acontecimiento deseable.

Expresiones: alegre, complacido, contento, encantado, eufórico, feliz, gozoso, jubiloso, sentirse bien, etc.

Variables que afectan la intensidad:

- El grado en que el acontecimiento es deseable.

Ejemplo: el hombre se puso contento al saber que había ganado la lotería.

Emociones de congoja

Especificación de tipo: (disgustado por) un acontecimiento indeseable.

Expresiones: acongojado, añoranza, apesadumbrado, conmovido, deprimido, desasosegado, descontento, desdichado, disgustado, duelo, enfadado, infeliz, solitario, triste, etc.

Variables que afectan la intensidad:

- El grado en que el acontecimiento es indeseable.

Ejemplo: El hombre se enfado cuando su computadora se descompuso repentinamente.

Emociones de estar feliz por

Especificación de tipo: (contento por) un acontecimiento que se supone deseable para otro.

Expresiones: contento por, encantado por, estar feliz por, etc.

Variables que afectan la intensidad:

- El grado en que el acontecimiento deseable para el otro es deseable para uno mismo.
- El grado en el que se presume que el acontecimiento es deseable para la otra persona.
- El grado en que la otra persona merece el acontecimiento.
- El grado de afecto hacia la otra persona.

Ejemplo: Alfredo estaba feliz por su amiga Sandra porque ella recibió un ascenso en su trabajo.

Emociones de lamentar por

Especificación de tipo: (descontento por) un acontecimiento presuntamente indeseable para alguna otra persona.

Expresiones: compasión, lástima, lamentar por, simpatía, triste por, etc.

Variables que afectan la intensidad:

- El grado en el que el acontecimiento indeseable para el otro es indeseable para uno mismo.
- El grado en el que el acontecimiento se presume indeseable para la otra persona.
- El grado en el que la otra persona no merecía el acontecimiento.
- El grado de afecto hacia la otra persona.

Ejemplo: Alfredo se compadeció de su amiga Sandra porque su perro preferido se perdió.

Emociones de resentimiento

Especificación de tipo: (descontento por) un acontecimiento que se supone deseable para alguna otra persona.

Expresiones: celos, envidia, resentimiento, etc.

Variables que afectan la intensidad:

- El grado en el que el acontecimiento deseable para la otra persona es indeseable para uno mismo.
- El grado en el que se presume que el acontecimiento es deseable para la otra persona.
- El grado en el que la otra persona no merecía el acontecimiento.
- El grado en el que no se tiene afecto por la otra persona.

Ejemplo: El ejecutivo tenía resentimiento por el aumento de sueldo otorgado a un colega al que consideraba incompetente.

Emociones de alegría por el mal ajeno

Especificación de tipo: (contento por) un acontecimiento que se presume indeseable para alguna otra persona.

Expresiones: alegría por el mal ajeno, schadenfreude (alemán), etc.

Variables que afectan la intensidad:

- El grado en el que el acontecimiento indeseable para la otra persona es deseable para uno mismo.
- El grado en el que se presume que el acontecimiento es indeseable para la otra persona.
- El grado en el que la otra persona merecía el acontecimiento.
- El grado en el que la otra persona no es apreciada.

Ejemplo: Los contrincantes del jugador se alegraron de su infortunio al verlo fallar el tiro.

Un grupo especial de *emociones son aquellas basadas en previsiones*, es decir las que involucran la previsión, confirmación o refutación de algún acontecimiento, donde la *variable de probabilidad* cobra especial atención. Estas emociones son:

Emociones de esperanza

Especificación de tipo: (contento por) la previsión de un acontecimiento deseable.

Expresiones: anticipación, esperanza, excitación, expectación, etc.

Variables que afectan la intensidad:

- El grado en el que el acontecimiento es deseable.
- La probabilidad del acontecimiento.

Ejemplo: Gabriela al pensar en la posibilidad de que le pidieran salir a bailar, se llenó de esperanza.

Emociones de miedo

Especificación de tipo: (descontento por) la previsión de un acontecimiento indeseable.

Expresiones: amedrentado, ansiedad, aprensión, aterrorizado, miedo, nervioso, pavor, petrificado, preocupado, etc.

Variables que afectan la intensidad:

- El grado en el que el acontecimiento es indeseable.
- La probabilidad del acontecimiento.

Ejemplo: Enrique, sabiendo que no había estudiado lo suficiente, tenía miedo de reprobado el examen.

Emociones de satisfacción

Especificación de tipo: (contento por) la confirmación de la previsión de un acontecimiento deseable.

Expresiones: complacencia, satisfacción, etc.

Variables que afectan la intensidad:

- La intensidad de la emoción concomitante de esperanza.
- El esfuerzo empleado en tratar de conseguir el acontecimiento.
- El grado en que se ha realizado el acontecimiento.

Ejemplo: Gabriela quedó complacida cuando se dio cuenta de que el muchacho de sus sueños la estaba invitando a salir.

Emociones de temores confirmados

Especificación de tipo: (descontento por) la confirmación de la previsión de un acontecimiento indeseable.

Expresiones: temores confirmados, etc.

Variables que afectan la intensidad:

- La intensidad de la emoción concomitante de miedo.
- El esfuerzo empleado en tratar de evitar el acontecimiento.
- El grado en que se ha realizado el acontecimiento.

Ejemplo: Los temores de Alfredo se confirmaron cuando supo que en su trabajo iba a ser transferido a una ciudad lejana.

Emociones de alivio

Especificación de tipo: (contento por) la refutación de la previsión de un acontecimiento indeseable.

Expresiones: alivio.

Variables que afectan la intensidad:

- La intensidad de la emoción concomitante de miedo.
- El esfuerzo empleado en intentar evitar el acontecimiento.
- El grado en el que se ha realizado el acontecimiento.

Ejemplo: La madre sintió alivio al saber que su hijo está fuera de peligro.

Emociones de decepción

Especificación de tipo: (descontento por) la refutación de la previsión de un acontecimiento deseable.

Expresiones: decepción, desesperanza, frustración, romper el corazón, etc.

Variables que afectan la intensidad:

- La intensidad de la emoción de esperanza concomitante.
- El esfuerzo empleado en intentar alcanzar el acontecimiento.
- El grado en el que se ha realizado el acontecimiento.

Ejemplo: David quedó decepcionado cuando se dio cuenta de que no podría visitar otro país.

La segunda categoría más importante se refiere a las *emociones de atribución*, es decir, aquellas ligadas a las acciones de los agentes. De este grupo se identifican cuatro emociones, dos de las cuales tiene la peculiaridad de que el agente sobre quien recae la responsabilidad de la acción es la misma persona que está experimentando la emoción.

Emociones de orgullo

Especificación de tipo: (aprobación de) una acción plausible de uno mismo.

Expresiones: orgullo

Variables que afectan la intensidad:

- El grado de la plausibilidad estimada.
- La fuerza de la unidad cognoscitiva con el agente real.
- Las desviaciones de la acción del agente de las expectativas basadas en la persona o el papel social (es decir, la cualidad de inesperado)

Ejemplo: La mujer estaba orgullosa de haber salvado la vida de un niño que se ahogaba.

Emociones de autorreproche

Especificación de tipo: (desaprobación de) una acción censurable de uno mismo.

Expresiones: autocondena, autorreproche, azoramiento, embarazoso, sentirse culpable, sentirse mortificado, vergüenza, etc.

Variables que afectan la intensidad:

- El grado de censurabilidad estimada.
- La fuerza de la unidad cognoscitiva con el agente real.
- Las desviaciones de la acción del agente de las expectativas basadas en la persona o el papel social.

Ejemplo: El jugador estrella se avergonzó de haber fallado el tiro decisivo en el partido de basketball.

Emociones de aprecio

Especificación de tipo: (aprobación de) la acción plausible de otra persona.

Expresiones: admiración, aprecio, consideración, estima, respeto, etc.

Variables que afectan la intensidad:

- El grado de la plausibilidad estimada.
- Las desviaciones de la acción del agente de las expectativas basadas en la persona o el papel social.

Ejemplo: Los colegas del físico lo admiraban por su obra, que había merecido el premio Nobel.

Emociones de reproche

Especificación de tipo: (desaprobación de) la acción censurable de otra persona.

Expresiones: desdén, desprecio, indignación, menospreciar, reproche, repugnancia, etc.

Variables que afectan la intensidad:

- El grado de la censurabilidad estimada.

- Desviaciones de la acción del agente de las expectativas basadas en la persona o en el papel social.

Ejemplo: Mucha gente despreció al político por haber engañado al país.

Al presentar las emociones de atribución, podemos considerar la valoración de un acontecimiento con base en la acción de un agente, sin embargo, este proceso no es excluyente de alguna otra valoración donde se esté considerando al acontecimiento mismo. Este hecho sugiere la posibilidad de que *dos emociones diferentes se presenten simultáneamente*, una de ellas basada en acontecimientos y la otra en emociones de atribución, ante la valoración de un acontecimiento desde distintos puntos de vista. Frente a esto, los autores de la teoría OCC consideran que existe un grupo de *emociones compuestas* que se resume en la siguiente tabla:

Aprobación de la acción plausible de otra persona (admiración)	+	Contento por un acontecimiento deseable (júbilo)	→	<i>Gratitud</i>
Desaprobación de la acción censurable de otra persona (reproche)	+	Disgusto por un acontecimiento indeseable (congoja)	→	<i>Ira</i>
Aprobación de una acción plausible de uno mismo (orgullo)	+	Contento por un acontecimiento deseable (júbilo)	→	<i>Complacencia</i>
Desaprobación de la acción censurable de uno mismo (vergüenza)	+	Disgusto por un acontecimiento indeseable (congoja)	→	<i>Remordimiento</i>

En el último grupo, las emoción de atracción se relacionan con reacciones momentáneas de agrado o desagrado hacia los objetos, éstas parecen ser más inmediatas, más espontaneas y menos afectadas por los procesos cognoscitivos. Se identifican únicamente dos emociones, que son:

Emociones de agrado

Especificación de tipo: (agrado por) un objeto atractivo.

Expresiones: adorar, afecto, atracción, gustar, etc.

Variables que afectan la intensidad:

- El grado en que el objeto es atractivo.
- El grado de familiaridad con el objeto.

Ejemplo: Gabriela se lleno de afecto cuando contemplo a su hijo recién nacido.

Emociones de desagrado

Especificación de tipo: (desagrado por) un objeto repulsivo.

Expresiones: aborrecer, aversión, desagrado, detestar, disgustar, odio, repugnancia, repulsión, etc.

Variables que afectan la intensidad:

- El grado en que el objeto es repulsivo.

- El grado de familiaridad con el objeto.

Ejemplo: La pareja sintió tanto desagrado por la película que se salió antes de que terminara.

La teoría OCC ha tenido una gran difusión y ha sido ampliamente usada por distintos modelos y arquitecturas en el área de la computación afectiva, esto es debido a que una de las metas planteada por sus creadores fue proporcionar un cimiento para un modelo que fuera tratable computacionalmente.

Como cualquier otra teoría, ésta no es perfecta, incluso sus autores son francos al indicar varias de sus debilidades. Al realizar un análisis con el objetivo de aplicar sus conceptos en un sistema computacional surgen diversos problemas, como la valoración relacionada a cada acontecimiento, aspectos que deben ser simplificados y acotados al dominio específico de forma que pueda ser implementado eficientemente.

2.3.2. El modelo EMA

Aunque la base de este trabajo se encuentra en lo propuesto por la teoría OCC, el modelo descrito en esta sección nos ofrece un ejemplo de otras investigaciones que incluyen el uso de emociones sobre el comportamiento de agentes en un ambiente virtual. Con lo que se expone la idea de que las emociones pueden ser utilizadas con éxito en sistemas con características cercanas a las de un videojuego.

La meta de la investigación presentada por Marsella y Gratch [GRAJ04] [MARS06] es crear un modelo computacional general de los mecanismos sobre los que se basan las emociones humanas, identificando a la emoción como un mecanismo que incluye un proceso de *valoración*. Este último tiene como función evaluar: 1) las circunstancias que rodean al individuo, y 2) la *actuación* o sea la respuesta generada por parte de esta misma evaluación.

Este modelo basa sus ideas en el trabajo propuesto por Smith y Lazarus [SMIC90], llamado sistema *cognitivo-motivacional-emotivo*, el cual es representativo de las teorías modernas de valoración y se muestra en la Figura 2.2.



Fig. 2.2 El sistema cognitivo-motivacional-emotivo

La *valoración*, según esta teoría, se define como la relación entre una persona con su ambiente físico y social, referido como *relación persona-ambiente*. La *actuación* determina como el individuo responde al resultado de la valoración de los eventos, dependiendo de esto se identifican las características más importantes para la relación persona-ambiente con lo que se calcula un potencial que será útil para mantener o cambiar éstas características. De esta forma se plantea que el comportamiento surge de una relación muy cercana entre la cognición, la emoción y las respuestas actuadoras. Inicialmente los procesos cognitivos sirven para construir una interpretación individual de cada uno de los eventos externos así como su relación con las metas y los deseos; la valoración toma esta interpretación en términos de un conjunto de características abstractas que son útiles para guiar el comportamiento, lo que en última instancia, la actuación, motiva acciones que cambien el ambiente (actuación centrada en el problema) o motivando cambios en la interpretación de esta relación (actuación centrada en la emoción). Todas estas respuestas parecen estar influenciadas por la manera en como el individuo valora la situación como amenazante o retadora. Con esta perspectiva podemos entonces lidiar con situaciones en las que bajo el mismo estímulo, un individuo actúa de dos maneras diferentes en distintos momentos.

Bajo estas ideas se creó un modelo denominado EMA, llamado así en honor del libro "*Emotion and Adaptation*", escrito por Richard Lazarus en 1991. EMA se basa en la teoría de valoración para construir agentes que sean capaces de establecer un diálogo interactivo orientado a tareas, control en tiempo real sobre el comportamiento verbal y no verbal, así como a responder adecuadamente a eventos externos presentes en su entorno.

EMA modela la valoración y la actuación a través de cinco etapas descritas a continuación:

1. Construir y mantener una interpretación causal de eventos provenientes del mundo en términos de creencias, planes e intenciones. Donde una interpretación causal es una representación explícita del actual estado mental de un agente, concerniente al pasado, presente y futuro de sus estados y acciones, además de probabilidad y conveniencia de las relaciones causales entre estas.
2. Genera múltiples cuadros de valoración que identifican las características de la interpretación causal en términos de las distintas variables que se consideren en el proceso de valoración.
3. Mapea cada cuadro de valoración individual en instancias individuales de emoción.
4. Conjunta todas las instancias emocionales en un estado emocional actual y estado de ánimo general.
5. Finalmente adopta una estrategia de actuación en respuesta al estado emocional actual, es decir, genera un comportamiento adecuado a la situación.

En [MARS04] [MARS01] se muestra una aplicación funcional de este modelo, el cual genera un ambiente virtual de entrenamiento militar, donde agentes inteligentes controlan a distintos personajes, llamados *humanos virtuales*, que representan a grupos locales, amigables y hostiles, así como otros miembros del equipo. Estos agentes cuentan con un esquema de colaboración, comunicación natural cara a cara con el usuario y con otros personajes virtuales, lo que requiere de una amplia integración de habilidades motrices, solución de problemas, expresiones faciales, expresiones corporales y lenguaje. Esta implementación del modelo EMA relaciona cada cuadro de valoración con una emoción de un tipo e intensidad específico. Las emociones están basadas en una representación inspirada en el trabajo propuesto por la *teoría OCC*.

2.3.3. Otras Arquitecturas y Modelos

En [PETP01] se presenta una discusión sobre la importancia de las emociones en los agentes, principalmente se menciona la importancia de mejorar la interacción entre los humanos y las computadoras, donde debe ponerse especial atención en no degradar la experiencia del usuario generando comportamientos desconcertantes y no naturales (credibilidad). Según este análisis, es poco probable que el comportamiento autónomo inteligente pueda ser alcanzado sin emociones, incluso la interpretación emocional puede ser un incentivo evolutivo hacia el desarrollo de capacidades cognitivas de mayor sofisticación. Esta discusión hace evidente la necesidad de generar modelos y arquitecturas que nos permitan aplicar esta característica a cualquier tipo de agente, sin embargo, encontramos que éste no es una empresa sencilla y que está llena de problemas de diseño, donde los agentes deben comportarse inteligentemente más que solo actuar con una casi perfecta racionalidad, sus movimientos deben comunicar toda la historia personal que está involucrada lo que además introduce la idea de que los agentes también deben mostrar una personalidad, lo que evidentemente es un gran problema en sí mismo.

Así nos encontramos con investigaciones como Allbeck et al. [ALLJ02] que inicia el camino para generar una implementación creando un esquema de representación de conocimiento llamado PAR (Parameterized Action Representation) el cual produce una abstracción simbólica de las acciones o instrucciones para un agente, especificando al agente mismo, los objetos relevantes, información de caminos, localización de lugares, formas de interacción, propósitos, etc. De esta forma el PAR puede usarse para representar los conceptos descritos por la *teoría OCC*, por ejemplo, la representación de un acontecimiento puede contener información relativa a los cambios que producen en el ambiente, las metas que afectan, los agentes involucrados, etc.

Una arquitectura propuesta por Bach et al. [BACJ06], llamada *MicroPsi*, extrae ideas presentadas por la teoría *Psi*, donde los agentes autónomos se basan la abstracción de sus metas como un conjunto de impulsos que los motivan a ejecutar comportamientos específicos. Además, sugiere el uso de un tipo de neurona artificial que se organiza en estructuras jerárquicas que identifican a los agentes. Los impulsos son divididos en tres diferentes categorías: los impulsos psicológicos (hambre, cansancio, etc.), impulsos sociales (afiliación o pertenencia a ciertos grupos) y los impulsos cognitivos (reducción de la duda y aumentar la competencia en alguna tarea). La ejecución y el alcance de estos impulsos podría sugerirnos la posibilidad de que exista una causa sobre la que pueda desencadenarse una emoción, aunque de hecho, esta teoría señala que las emociones son solo una respuesta emergente al modelo modular de percepción, comportamiento y procesos cognitivos, es decir, las emociones solo son consideradas como una configuración específica de parámetros y no como

un elemento individual que requiere de sus propios procesos para ser significativo para el agente.

Una aproximación más general la encontramos en Wilson [WIL99], donde su motivación es proporcionar la información afectiva al desarrollador de esta forma determinar qué hará el agente para alterar su entorno. Esta arquitectura divide el concepto de las emociones en tres etapas: las *emociones momentáneas*, que se encuentran en el nivel más alto, son aquellas que se activan brevemente ante la reacción a eventos; el *estado de humor* es un estado emocional más prolongado, causado por el efecto acumulativo de las emociones momentáneas; por último en la capa más inferior se encuentra la *personalidad*, que generalmente es expuesta al no estar sobrescrita por alguna capa superior. Sin embargo, su desventaja está en el limitado grupo de emociones que considera, que son: alegría, tristeza, miedo, enojo, sorpresa e indignación. Para superar este problema, algunos investigadores prefieren guiar sus esfuerzos por las teorías de valoración que les permiten representar un rango más amplio del espectro emocional, un ejemplo de esto lo encontramos de Chown et al. [CHOE02] que concentra sus objetivos en demostrar como las emociones pueden afectar la toma de decisiones y como los efectos pueden variar de acuerdo a la personalidad del agente. Otra arquitectura [MALM04] enfocada a robots autónomos basa directamente su definición de emociones en *la teoría OCC*, sin embargo, su uso es más discreto pues considera que existen comportamientos no emocionales, es decir, que no son afectados por ningún estado afectivo. De esta forma si considera que no existe una emoción activa, un control dominante selecciona la meta del robot de forma lógica, mientras que si se presenta una emoción entonces esta tendrá influencia en el módulo de decisión.

Finalmente vale la pena mencionar el proyecto *SAFIRA* (Supporting Affective Interactions for Real-time Applications), propuesto por Paiva et al. [PAIA08], un esfuerzo por crear una plataforma que permita crear aplicaciones interactivas enriquecidas por el aspecto emotivo, proporcionando un conjunto de herramientas para tareas como la adquisición de conocimiento afectivo, representación, razonamiento, planeación, comunicación y expresión. La investigación realizada en *SAFIRA* busca incluir las tres fases fundamentales de la computación afectiva: 1) el registro del estado afectivo con ayuda principalmente de objetos externos que permitan interpretar la entrada del usuario al sistema, 2) formar un modelo de su estado afectivo; el razonamiento afectivo donde se incorporan y combinan los componentes de cognición así como el de emoción en un agente; y 3) la expresión afectiva donde se desarrollan técnicas de comunicación más creíbles.

Los resultados hasta el momento del proyecto *SAFIRA* le han permitido desarrollar tres demostraciones conceptuales:

James the Butler, opera dentro de una aplicación de ventas en línea de una vinatería. James es un *Asistente Personal de Ventas* (Personal Sales Assistant o PSA) que utiliza métodos de razonamiento adaptativo, para proporcionar una asistencia colaborativa al cliente. El objetivo de esta aplicación es evaluar el uso de agentes autónomos con las habilidades necesarias para mostrar un comportamiento visual y verbal, similares a las de un vendedor humano.

FantasyA es un videojuego donde los usuarios juegan el rol de un aprendiz de hechicería quien es retado a encontrar al líder de su clan a través de la exploración, duelos y cooperación. El objetivo para el desarrollo de este juego fue el uso de los componentes de SAFIRA que permiten la entrada de datos afectivos por medio de una interfaz física llamada *SenToy*, y la expresión de las emociones por medio de los movimientos corporales de los personajes.

La tercera aplicación es llamada *The Influencing Machine*, que no es más que una instalación interactiva donde las personas pueden tomar una tarjeta e introducirla en la máquina influenciando de esta manera las emociones de un agente que se expresa a través dibujos generados en tiempo real, similares a los dibujos de un niño pequeño.

2.4. En nuestro caso de estudio

A lo largo de este capítulo hemos introducido los términos y teorías sobre los que se fundamenta nuestra propuesta del uso de las emociones. Aunque la definición de emoción sigue siendo un debate actual en la rama de Psicología, hemos planteado una descripción aproximada que nos permitirá definir los conceptos claves sobre los que se fundamenta nuestra arquitectura. Basados en la evidencia propuesta, asimilamos la idea de que cualquier comportamiento generado a partir de un proceso cognitivo previo es influenciado por las emociones activas en ese momento.

Apoyándonos en la discusión sobre la computación afectiva, identificamos a nuestro proyecto en términos de la creación de emociones artificiales que afecten el comportamiento de nuestros agentes con el objetivo de dotar su comportamiento de una mayor credibilidad al escrutinio humano con el apoyo de otras técnicas como la animación y la expresión facial. Para lograr esto, definimos nuestro rango de emociones basados en la *teoría OCC*, que además nos ofrece conceptos útiles para representar las metas de los agentes, así como otros medios para realizar una valoración del entorno, indispensable para identificar las condiciones necesarias para que nuestros agentes experimenten una emoción.

Finalmente en este capítulo se presentaron, algunos de los modelos y arquitecturas revisados a lo largo de esta investigación, los cuales nos

permitieron identificar algunas de las características que presentan las implementaciones existentes y las diferentes aproximaciones que estas usaron.



Capítulo 3

Videojuegos Afectivos

3.1. Estado del arte

Los videojuegos pueden ser analizados desde un gran número de facetas, las cuales, son realmente distintas entre si. Actualmente, la característica más detallada de un videojuego puede ser la capacidad visual que éste presenta, aunque existen muchos otros aspectos a considerar, como el diseño del juego, la historia, la originalidad, el factor de diversión, altos niveles técnicos en diferentes áreas como el sonido y la animación, etc. Como se mencionó en el capítulo uno del presente trabajo, el aspecto gráfico ha evolucionado con los años hasta alcanzar un nivel técnico muy elevado, y aunque no todo juego toma provecho de estas capacidades, si existe una tendencia de algunos por crear entornos con personajes realistas. Instintivamente, si tuviéramos una imagen estática de alguno de estos juegos, como los mostrados en la Figura 3.1, podemos imaginar una historia detrás de lo que vemos, cómo llegaron a dicha situación, que podría estar motivando al personaje; cualquier pista podría entonces guiarnos hacia un contexto de lo que sucede, debido a su calidad visual la inmersión en cualquiera de las situaciones será aún mayor pues en algunos casos podríamos creer que son imágenes reales y no generadas por computadora. Pero, ¿qué relación existe entre estos videojuegos y las emociones?



Fig. 3.1. *Army of Two* de Electronic Arts (arriba) y *Gears of War 2* de Epic Games (abajo)

Si tomamos por ejemplo una novela como *El viejo y el mar* (escrita por Ernest Hemingway en 1951) encontramos un conjunto de elementos que ayudan a evocar nuestras emociones por medio de la descripción de los personajes y los sucesos que ocurren alrededor de ellos. Cualquiera que haya leído un buen libro puede ser testigo de este fenómeno que no es único del género pues otros medios como la televisión y las películas provocan un estímulo similar, apoyados por elementos propios como el manejo de cámara y efectos sonoros. Entonces como otro medio más, un videojuego tiene también esta misma capacidad latente para impactar en las emociones de los jugadores, pero como lo demuestra el primer ejemplo, esto es posible aún sin el potencial gráfico que las técnicas y hardware actual pueden ofrecer.

Haciendo referencia a los inicios de los videojuegos, encontramos juegos como *Pong* (1972) o *Tetris* (1985), los cuales por su sencilla temática, esencialmente a causa de la capacidad técnica, consistían en seguir un grupo de reglas para continuar el juego e incrementar una puntuación. Esto, evidentemente, no incluía ningún factor emocional, sin embargo, la competitividad y el deseo de superación por alcanzar una puntuación más alta podía provocar en los jugadores intensos episodios emotivos, similares a los ocasionados en otros eventos como un encuentro deportivo. En esa misma época, sobresalían algunos otros juegos como *Space Invaders* (1978), *Pac-Man* (1979) y *Donkey*

Kong (1981) donde se ofrecía una idea muy básica de la situación con unos cuantos personajes, sobre la cual las personas podían especular sobre alguna historia detrás de lo que veían. Conforme los juegos iban evolucionando, los desarrolladores tenían la capacidad de mostrar un enfoque distinto capaz de involucrar con mayor éxito al jugador. El videojuego *Punch-out!!* (1987) permitía tomar el rol de un boxeador amateur, cosechando victorias y campeonatos, permitiendo al jugador enorgullecerse de cada pelea, donde en cada transición recibíamos un incentivo para seguir adelante, ya sea con palabras del entrenador u observando el daño a la cara que hubiera recibido cualquiera de los contendientes, además junto con el diseño y personalidad dada a cada boxeador eran ingredientes suficientes para evocar algunas emociones intensas en el jugador. En ese mismo año la compañía *Square-Enix* [SQUE08] presenta el juego *Final Fantasy*, que es considerado uno de los primeros juegos tipo RPG², donde apoyados por simples líneas de texto presentaban una historia compleja que se iba desarrollando conforme se cumplían objetivos dentro del propio juego. De manera similar a un cuento escrito en un libro, *Final Fantasy* trataba de involucrar al jugador evocando sus sentimientos desde un inicio mostrando una pantalla inicial con el siguiente texto, justo después de encender la consola:

*The world is veiled in
darkness. The wind stops,
the sea is wild,
and the earth begins to rot.
The people wait,
their only hope, a prophecy...*

*'When the world is in darkness
Four Warriors will come...'*

*After a long journey, four
young warriors arrive,
each holding an ORB.*

De esta forma, poco a poco iban evolucionando diferentes elementos que buscaban influir directamente en las emociones de los jugadores. Sin embargo, debido a sus diferencias con otros medios como las películas, los videojuegos no solo se valen de algunos recursos ya probados, sino que también deben adaptarse a sus características. Federico Peinado [PEIF05] identifica aspectos relacionados a lo que él considera *narraciones interactivas*, donde la ausencia

² *Role Playing Game* o simplemente *RPG* es un género donde el jugador toma el rol de un personaje para guiarlo, tomando decisiones importantes para sus objetivos e incrementando las características que lo identifican paulatinamente. Estos juegos se caracterizan por tener historias largas y envolventes, donde generalmente existen objetivos principales y secundarios no tan importantes para el desarrollo de la trama o el juego.

de historia no es un factor determinante, pues la gente tiende a crear la suya propia, pero cuando ésta existe debe envolver e interesar al usuario. De hecho Peinado nos señala que los personajes presentes en un videojuego deben tener algo especial en ellos, ya sea que sean interesantes por sí mismos, o puedan hacer algo sobresaliente y que el secreto es ofrecer personajes creíbles, lo cual puede ser alcanzado de muchas maneras distintas, una de las cuales es el objetivo de la presente investigación.

Conforme el hardware y el software lo permitían, los videojuegos incluían elementos cada vez más complejos que incluían no solo texto sino diálogos grabados y video de alta resolución, lo cual ha sido explotado grandemente mostrando rasgos que solo eran vistos en el área cinematográfica. De esta forma cada personaje en un videojuego puede ser realzado con una enorme carga emocional, guiada por el contexto original y presentada por artistas de distintas áreas. Incluso en [CHAA03] se identifican un conjunto de razones de porqué buscamos que estos personajes muestren su lado afectivo y se presentan distintas formas de hacerlo que incluyen: uso de secuencias de video, relación de objetos en la historia con objetos del juego, atención al comportamiento de personajes autónomos, entre otros. Desde un punto de vista diferente la empresa *XEODesign* [XEOD08] realizó una investigación [LAZN04] sobre el factor emotivo que nos incita a seguir jugando, en la cual, identificaron cuatro puntos clave, siendo cada una la razón de porqué jugamos y un mecanismo para manipular las emociones en diferentes aspectos de la experiencia del juego. Estos puntos son:

1. *Diversión con Reto (Hard Fun)*. Los jugadores gustan de oportunidades para ser retados, el pensamiento estratégico y la resolución de problemas. Con frecuencia se generan emociones y experiencias como la frustración o el triunfo personal.
2. *Diversión con Facilidad (Easy Fun)*. Los jugadores disfrutan de la intriga y la curiosidad. Experimentan una profunda inmersión en el juego, cuando éste absorbe su atención. Las características de inmersión producen emociones y experiencias de asombro, admiración, miedo además de misterio.
3. *Estados Alterados (Altered States)*. Los jugadores disfrutan de sus experiencias internas en reacción a propiedades sociales, cognitivas o de comportamientos. Las personas que juegan por un estado alterado experimentan sensaciones como excitación o alivio de sus pensamientos y emociones.
4. *Las Personas (The People Factor)*. Los jugadores usan a los videojuegos como un mecanismo de experiencias sociales. Disfrutan de la diversión, schadenfreude (alegría por el mal ajeno) y el orgullo, experimentados por una relación social como la competencia o el trabajo en equipo que

además les permite crear lazos sociales, así como el reconocimiento personal frente a otros.

Así la industria de los videojuegos ha llegado a un punto donde encontramos juegos con cargas emocionales muy intensas. Por ejemplo, la desarrolladora *LionHead Studios* [LIOH08] ha puesto a la venta su último juego llamado *Fable II* (2008) en el cual se remarca la importancia de las emociones concentrando el enfoque afectivo a un agente autónomo, personificado por un canino, que acompaña al jugador a lo largo del desarrollo del juego, el cual se basa en las decisiones a problemas morales y sus consecuencias. Incluso juegos como *Façade* [FACA08], *Pearl Demon* [PERD08] y *Mr. Bubb in Space* [MRBS08] poseen un enfoque muy directo a las emociones, y remarca cómo éstas pueden ser experimentadas por personajes dentro del juego.

Los videojuegos cada vez son más complejos y con cada generación la industria muestra un nuevo avance en cada área, donde las emociones no son la excepción, ya sea que un personaje posea una expresión facial para cada estado que experimente o una historia realmente envolvente, cada nuevo juego tiene el poder de hacer surgir nuestras emociones de distintas formas, lo que puede ser un factor para su éxito en el mercado o en el recuerdo de las personas que lo jugaron.

3.2. Componentes de un videojuego

Como cualquier otro sistema, un videojuego se divide en subsistemas para manejar la complejidad creciente conforme aumentan sus características. Un *Game Engine* o *Motor de Juego* es un conjunto de componentes especializados que llevan a cabo diferentes tareas necesarias para cada fase del ciclo que forma un videojuego el cual, básicamente, consta de recibir las señales de entrada de parte del usuario, procesar la lógica interna y dibujar en pantalla una imagen representativa del estado del juego, este ciclo se repite hasta que ciertas condiciones se cumplan.

Los primeros videojuegos y aquellos relativamente simples raramente usan un *Game Engine* como base, sin embargo, los videojuegos con más características manejan esta complejidad repartiendo los procesos en distintos componentes. De esta forma, estos sistemas pueden estar formados por uno o más de los componentes descritos a continuación.

3.2.1. Servicios de IO

Generalmente, una aplicación multimedia como un videojuego posee un conjunto de objetos que empaquetan gran cantidad de información y recursos

útiles a lo largo de aplicación, estos objetos son llamados *assets*, los cuales pueden variar desde imágenes estáticas, secuencias de video, efectos de sonido, bandas sonoras, modelos tridimensionales, secuencias de animación, guiones, etc. Para que cualquier aplicación pueda hacer uso de estos *assets*, es necesario que éstos sean recuperados de sistema de almacenaje donde estos se encuentran. Los servicios de IO, es un componente que en un videojuego se encarga de leer y cargar a la memoria principal todos aquellos *assets* necesarios para el presente estado del juego. Por ejemplo, para mostrar la pantalla inicial el sistema probablemente obtenga los *assets* que representan la imagen del fondo, la música que se reproducirá y el texto que se mostrará en el menú principal. Además si existe la posibilidad de leer de un dispositivo de almacenamiento, también debe existir soporte para la operación de escritura, la cual usualmente es utilizada para guardar información relativa al usuario y su avance en el juego.

El segundo aspecto relativo a los servicios de IO y tal vez el más importante, es el relacionado con recibir la información proveniente del usuario, es decir, el jugador a través de algún tipo de dispositivo de entrada (teclado y joystick, que es lo más común) envía señales que el sistema procesa e interpreta como acciones que afectan el estado actual del juego.

3.2.2. Motor de Render

Tal vez el componente más mencionado y reconocido en un videojuego es el *Motor de Render* o *Motor de Dibujo*, el cual también es el más desarrollado en cuanto a tecnologías para mejorar su calidad y rendimiento con algoritmos más eficientes. Este componente se encarga de tomar ciertos datos específicos del juego en cada ciclo para generar una imagen representativa del estado actual.

Aunque muchos juegos tienen una representación bidimensional, en la actualidad un motor de render se especializa en dibujar objetos tridimensionales en una pantalla en dos dimensiones. Con lo cual un componente de esta clase tiene a su cargo las siguientes responsabilidades [EBED06]:

- Transformar la información 3D de los objetos, basándose en un modelo de cámara, posteriormente, convierte estos datos en información 2D, la cual podrá ser dibujada como píxeles en una pantalla.
- Eliminar porciones de los datos que no son visibles al observador. Esto involucra conceptos como *culling* (proceso que determina si un objeto está completamente fuera de la vista) y *clipping* (proceso que divide un objeto en partes más pequeñas, donde las partes no visibles serán descartadas y las visibles serán procesadas posteriormente).
- Dibujar toda la información 2D obtenida en procesos previos. Este proceso es llamado *rasterización*.

Los desarrolladores cuentan con bibliotecas gráficas que les permiten realizar todas estas tareas para crear su propio motor de render, como *DirectX* [DIRX08] y *OpenGL* [OPEG08], que son enriquecidos por lenguajes de bajo nivel como son *HLSL* (High Level Shading Language) y *GLSL* (OpenGL Shading Language) todo esto apoyado por hardware especializado [NVID08] [ATIA08] para lograr la mejor calidad de imagen.

3.2.3. Motor de física y detección de colisiones

Una de las tareas más comunes que conciernen a los videojuegos y ambientes virtuales es la detección de colisiones que consiste en detectar cuando dos objetos diferentes ocupan total o parcialmente el mismo espacio. Adicionalmente, dado que existen fuerzas involucradas en el movimiento, estas también pueden ser calculadas para modificar la posición de los objetos generando escenas de gran realismo debido a su comportamiento basado en leyes físicas extraídas de nuestro propio mundo, las que pueden incluir a la gravedad, fuerzas cinéticas, inercia, fricción, etc.

Debido a la gran cantidad de operaciones que este componente potencialmente puede realizar existe también hardware especializado [PHYN08] útil para los juegos más demandantes.

3.2.4. Administrador Jerárquico de escena

El motor de render mencionado anteriormente requiere que cada objeto con el potencial de ser dibujado, pase ciertas pruebas de visibilidad. De igual manera el motor de física toma los objetos en el mundo virtual y realiza pruebas a cada uno de ellos para detectar una colisión. En un mundo con un gran número de objetos el método más sencillo es procesar todos los objetos agrupándolos en una lista e iterando sobre cada elemento de ésta. Sin embargo, aunque esta aproximación es simple es también muy ineficiente. Un mejor método para procesar los objetos es agrupándolos en una estructura jerárquica de acuerdo a su posición espacial, tipo o algún otro parámetro. Esta organización es llamado *administrador jerárquico de escena* o simplemente *scenegraph* la que usualmente es implementada por medio de una estructura tipo árbol.

La importancia de un *scenegraph* en los videojuegos radica en su potencial para reducir una gran cantidad de procesos en distintos componentes, por ejemplo, al no procesar la animación de personajes fuera de la vista del jugador, o evitar cálculos de fuerzas a objetos muy lejanos al origen de una explosión cuando estos son casi imperceptibles.

3.2.5. Servidor de sonido

El servidor de sonido tiene a su cargo las tareas relativas principalmente a la reproducción de sonido, lo que involucra música, efectos sonoros, diálogos y la sincronización de estos con otros elementos. Es también encargado de producir efectos relativos a la física de las ondas sonoras como oclusiones, obstrucciones, características de distintos materiales, posición y dirección en un ambiente tridimensional.

3.2.6. Ejecutor de guiones

Un ejecutor de guiones es el componente que interpreta un *guión (script)* escrito en algún *lenguaje de scripting* como LUA [LUA08], UnrealScript [SWET98] o Python [PYTH08], con el objetivo de controlar la aplicación y ciertos componentes de esta. En la industria de los videojuegos es común el uso de scripts pues permiten el desarrollo de ciertos componentes o características sin afectar directamente el código principal de la aplicación, incluso esta separación llega hasta el usuario, el cual es capaz de modificar o agregar funcionalidad al juego únicamente con la creación de nuevos scripts. Algunas de las tareas más comunes realizadas en videojuegos a través de lenguajes de scripting y el ejecutor de guiones son: procesos de inicialización, manejo de lógica del juego, tareas de inteligencia artificial, control de cámara, manipulación de eventos, generación de secuencias de animación, etc.

En frecuentes ocasiones la importancia de un ejecutor de guiones radica en la complejidad de los sistemas, cuanto más grandes se hacen los proyectos más gente debe involucrarse en el desarrollo, esto es de vital importancia para la Inteligencia Artificial en videojuegos debido a que los desarrolladores pueden crear guiones que contengan las conductas que los personajes mostrarán, aún cuando el producto final siga en desarrollo por otro equipo diferente.

3.2.7. Servidor de animación

Un servidor de animación lleva a cabo procesos relativos a los cambios en los parámetros de los objetos con respecto al tiempo. Existen distintas técnicas con las cuales este componente tiene que lidiar dependiendo del tipo de objeto se este animando, ya sea objetos sólidos que se mueven a una velocidad constante, el complejo movimiento de un personaje cuadrúpedo corriendo a distintas velocidades o la interacción entre distintos objetos suaves como la tela.

Técnicas comúnmente usadas como animación por cuadros clave (key frame), animación por esqueleto (skinning), interpolación, cuerpos rígidos (rigid body),

cuerpos blandos (soft body), cinemática inverso y directa son extensamente descritos en [PARR02] y [WATA92]

3.2.8. Servicios de red

Algunos videojuegos tienen la característica de permitir que dos o más jugadores interactúen en forma simultánea a través de una red de computadoras. De esta forma usando internet, jugadores localizados en distintas partes del mundo pueden interactuar en el mismo entorno virtual. Para lograr esto un componente dedicado debe proporcionar distintos servicios que permitan una constante comunicación entre los sistemas para el envío y recepción de datos.

Los servicios de red generalmente están contruidos bajo uno de dos tipos de arquitecturas diferentes: Punto a punto (Peer to peer) y Arquitectura Cliente/Servidor. Cada una cuenta con un conjunto de protocolos (generalmente UDP y TCP/IP) que permiten intercambiar los datos de manera ordenada y segura.

Uno de los retos que un sistema de las características de un videojuego debe enfrentar en relación con los servicios de red es el llamado *Lag*, que es el retraso o pérdida de algunos de los datos transmitidos por los servicios de red. En el caso de un videojuego el lag puede experimentarse al recibir los datos de otro jugador que indican su posición espacial en el entorno virtual, lo cual podría ocasionar que el jugador este recibiendo esta información en donde perciba a otros moviéndose erróneamente o desapareciendo de un punto para reaparecer en otro. El sistema, en estos casos debe utilizar técnicas de predicción que minimicen estos efectos dando a los jugadores una experiencia lo más fluida posible bajo las condiciones de red presentes.

3.2.9. Motor de Inteligencia Artificial

Para algunos juegos el motor de Inteligencia Artificial, o simplemente motor de IA, es indispensable para controlar los comportamientos y acciones de agentes autónomos presentes en el juego. Este es el componente sobre el cual se enfoca este trabajo por lo que se proporciona en las siguientes secciones una descripción más detallada de sus responsabilidades y técnicas usadas.

3.3. La inteligencia artificial en los videojuegos

3.3.1. Descripción

Una de las áreas de mayor investigación en las ciencias computacionales es la *inteligencia artificial (IA)*, en la cual cada una de sus subáreas poseen por sí mismas un gran reto a sus propias metas. Nuestro objetivo no es definir la IA, ni exponer ninguno de sus paradigmas, esta información es expuesta de mejor manera en [RUSS04], sin embargo, debemos hacer una diferencia entre ésta inteligencia artificial *clásica* (también llamada de academia) y la inteligencia artificial en los videojuegos (*Game AI*).

Según Brian Schwab [SCHB04] la inteligencia artificial en los videojuegos se define como:

El código en un juego que hace que los oponentes controlados por la computadora (o elementos cooperativos) parezcan tomar decisiones inteligentes cuando el juego tiene múltiples opciones para una situación dada, resultando en comportamientos que son pertinentes, efectivos y útiles.

Por otro lado, Nayarek [NARA04] nos recuerda que la IA en videojuegos, no tiene como meta calcular el comportamiento óptimo que venza al jugador, sino que el resultado debe reflejarse en comportamientos y acciones creíbles que sean lo más divertidas posibles. Así podemos enfatizar que la IA controla a los agentes o NPC³ para que estos interactúen dentro del juego encontrando un equilibrio entre la búsqueda de una solución óptima para un problema y la diversión que pueda proporcionar al jugador mostrando características que se esperaría encontrar en un individuo inteligente. Para cumplir estos objetivos puede ayudarse de técnicas y algoritmos descritos por la IA clásica, o de cualquier otro método que le permita cubrir los requerimientos y restricciones de aplicaciones de visualización en tiempo real como lo es un videojuego.

En los videojuegos, la IA tiene a su cargo distintas tareas como análisis topológico, búsqueda de caminos, evasión de obstáculos, movimiento de multitudes, coordinación de equipos, administración de sistemas perceptivos, toma de decisiones, etc., muchos de estos tomados en cuenta por motores de IA comerciales como Autodesk Kynapse [AUTK08] y Presagis AI.implant [PREA08]. Estos motores proporcionan herramientas para desarrollar las tareas más comunes, sin embargo, aún con estos productos a su disposición los desarrolladores se enfrentan con una ardua tarea de maximizar los recursos disponibles. Para lo cual utilizan cualquier truco con el fin de incrementar el realismo y la diversión. Sin embargo, aún no es posible establecer que en la industria exista un nivel aceptable de IA en los agentes, aunque algunos

³ NPC o *Non Player Character* se refiere a cualquier personaje dentro del juego que no este siendo controlado por un jugador humano y por tanto debe ser controlado por el sistema.

videojuegos muestran un admirable avance [CHAA07] aún encontramos situaciones en las que claramente el jugador pierde la sensación de inmersión al ser acompañados por personajes *acartonados*, es decir, que parecen mostrar un comportamiento muy lineal. Aún dada esta situación Nayarek [NARA07] enfatiza que no hay nada malo con la IA tradicional en videojuegos, pero se requiere de una IA a un nivel más alto que utilice estas técnicas como sus cimientos.

3.3.2. Percepción

En [LONS95] se define la *percepción* como un proceso en el cual recibimos e interpretamos información del mundo a nuestro alrededor, que además es afectada por el conocimiento, las emociones y la motivación. Así, en un sistema de IA la percepción, se refiere a los procesos por los cuales un agente adquiere o recibe información relativa al mundo virtual, otros agentes y su estado mismo. Las capacidades de percepción de un agente o un NPC en un sistema como lo es un videojuego puede variar desde conocer únicamente la posición del jugador, hasta un amplio conocimiento de las capacidades de ataque de cada enemigo que se ha presentado a lo largo del juego. Además puede variar en diferentes aspectos como: la frecuencia con la que se reciben los datos, la capacidad para retenerlos, su sensibilidad espacial, etc. Es importante un adecuado diseño del sistema perceptivo de los agentes, éstos generalmente deben tener las mismas capacidades de percepción que el jugador, de lo contrario podría existir una considerable desventaja que en ocasiones, cuando ésta es evidente, rompe con la ilusión de inteligencia [SCOB02].

La percepción externa se refiere a los sucesos que existen en el mundo virtual fuera del propio agente. Generalmente, en agentes con características humanoides la percepción externa es concebida emulando los sentidos humanos, es decir, el sentido de la vista, oído, olfato, gusto y tacto. Sin embargo, debido a la contextualización de cada juego, los sentidos más usados son la vista y el oído.

La percepción interna, la cual también llamamos propiocepción⁴, proporciona al agente información sobre él mismo, en otras palabras su estado interno, deseos y motivaciones. En agentes que simulan seres vivos, ésta información puede indicarles su necesidad de comida, sueño, cansancio, o *estados afectivos*; datos que normalmente no son expuestos directamente a otros agentes. En otros agentes la propiocepción puede suministrar datos relativos a su posición, orientación, armamento, energía interna, etc.

⁴ Para una descripción más extensa de la propiocepción referirse a [GANS93]

3.3.3. Selección de acciones

El sistema para la toma de decisiones es el bloque principal que determina la arquitectura del motor de IA. La inferencia [SCHB04] se define como el acto de derivar conclusiones lógicas y razonables de conocimiento factual o premisas que se presumen ciertas. Lo que en términos de un entorno virtual significa que cualquier agente controlado por el motor de IA obtiene información del mundo por medio de la percepción y toma decisiones razonables e inteligentes en relación a la acción que se da como respuesta.

El sistema de inferencia está limitado al tipo y la cantidad de información que este posea, así mientras más cosas el juego le permita a los agentes hacer, mayor será el conjunto de respuestas, o *espacio de estados*, que éstos puedan utilizar. En dependencia del tamaño de este espacio de estados existen distintas técnicas que el sistema de IA puede utilizar para realizar el proceso de inferencia, algunas de las cuales serán expuestas en la *sección 4* de este mismo capítulo.

En general, existen dos tipos de soluciones a los problemas que puede enfrentar un agente: *estratégicas* y *tácticas*. Una solución estratégica usualmente es vista a largo plazo para satisfacer las metas de más alto nivel lo que puede involucrar un conjunto amplio de acciones para su conclusión; por otro lado, una solución táctica se relaciona con metas de bajo nivel que pueden ser alcanzadas a corto plazo, y que usualmente involucran la ejecución de una acción o habilidad específica. Esta división puede ejemplificarse claramente, si pensamos en un juego bajo un contexto militar en donde se encuentra una solución estratégica que indica que debemos mover un grupo de soldados desde el campo de batalla hacia la parte vulnerable de una ciudadela para evitar un ataque sorpresa; sin embargo, distintas soluciones tácticas pueden presentarse en la acción de trasladar a los soldados hacia el punto destino pues deberán encontrar un camino cercano al óptimo y actuar de forma coherente en caso de encontrar enemigos en el trayecto.

3.3.4. Movimiento autónomo

Cuando hablamos de un agente con movimiento autónomo, usualmente nos referimos a la acción de trasladarse en su entorno y responder adecuadamente a una situación inesperada para ajustar su movimiento. La *navegación* es el problema de trasladar un agente desde un punto A, hasta un punto B dentro del entorno. Este ha sido un problema ampliamente estudiado por el área de la robótica y de optimización.

La navegación involucra dos niveles de análisis, dependiendo de la complejidad del movimiento que se requiera. En un nivel bajo, técnicas como los *campos potenciales* [PADC08] guían al agente por medio de suma de fuerzas atractivas

hacia el punto destino y fuerzas repulsivas lo alejan de obstáculos conocidos. Por otro lado, los *comportamientos dirigidos* (*steering behaviors*) [BUCM05] utilizan una sumatoria de fuerzas cuya resultante indica la dirección del movimiento, lo que permite una diversidad de movimientos como:

- *Buscar*. Produce una fuerza que dirige al agente hacia su destino.
- *Llegar*. Produce una fuerza hacia el destino que va disminuyendo conforme se acerca a éste de forma que el agente pueda detenerse grácilmente.
- *Huir*. Produce una fuerza en sentido opuesto a la dirección del destino.
- *Perseguir*. Produce una fuerza dirigida hacia una posición de intersección predicha por el movimiento de otro agente.
- *Evadir*. Produce una fuerza en sentido contrario a la dirección predicha por la posición de intersección con otro agente en movimiento.
- *Deambular*. Produce una fuerza que simula el movimiento aleatorio de un agente en su entorno.

En lo que podríamos considerar un nivel más alto en el movimiento autónomo, un agente puede usar *mapas de navegación* para trazar una ruta óptima que lo lleve de un punto a otro. La *planeación de rutas* (*pathplanning*) [BUCM05] [HIGD02] es una técnica que utiliza este mapa de navegación, que generalmente es implementado con un grafo donde cada nodo representa una posición espacial dentro del entorno, y por medio de algoritmos como Dijkstra y A* [STOB02] [MATJ02] se encuentra el conjunto de nodos que representan la mejor ruta a seguir desde la posición actual hacia el punto destino. La planificación de rutas perfecta sigue siendo un punto árido en el diseño de videojuegos, y de optimización. Existen gran cantidad de algoritmos de optimización que pueden desarrollarse. Sin embargo debe tenerse especial cuidado de no malgastar los escasos recursos del sistema asignados al motor de IA. El movimiento autónomo convincente es una parte crucial para dotar de cualquier agente virtual de un nivel de inteligencia aceptable, pues cualquier agente que no pueda evadir una pared con facilidad o tome el camino más largo y peligroso para llegar a su destino indudablemente nos hace dudar sobre su capacidad intelectual, lo que inevitablemente no causará una buena impresión hacia el jugador.

3.4. Técnicas utilizadas en los videojuegos

Esta sección presenta algunas de las técnicas más utilizadas por los sistemas de IA en videojuegos. Cada una presenta ventajas y desventajas de su implementación, también han demostrado su utilidad para dotar de un nivel de inteligencia a los agentes permitiéndoles ser un reto y diversión para el jugador.

3.4.1. Patrones

En sus inicios, antes de que la IA en videojuegos se consolidara como una parte necesaria, la mayoría de los agentes basaban su comportamiento en un *patrón*. Dicho patrón se formaba por una secuencia de movimientos y acciones bien definidos que se repetían en un ciclo infinito. Juegos como *Space Invaders* (1978) es un claro ejemplo de esta técnica donde los *invasores* se acercaban a la tierra en un movimiento que claramente seguía un patrón establecido.

Obviamente, el uso de patrones era fácilmente derrotado por un jugador atento lo que condujo a los desarrolladores a incluir otros aspectos, como agregar puntos débiles, lo cual aumentaba el reto y ayudaba a la creencia de contar con un enemigo más inteligente, debido a la dificultad para poder ser derrotado.

3.4.2. Máquinas de Estados Finitos

Las *máquinas de estados finitos* o *FSM* por sus siglas en inglés, han sido un modelo ampliamente utilizado en el diseño de circuitos, según [SAVJ04] contiene los elementos necesarios para describir la conducta de un sistema en términos de entradas, salidas, tomando en cuenta el tiempo. En [SCHB04] se define una FSM como una estructura de datos que contiene tres elementos: 1) todos los estados pertenecientes a esta máquina, 2) un conjunto de condiciones de entrada, y 3) funciones de transición que sirven como conexión entre los diferentes estados.

Usualmente, al modelar un comportamiento usando FSM, se diseña un esquema donde cada estado encapsula toda la lógica relacionada y que el agente debe ejecutar, esto ayuda en el diseño modular del sistema. Lo anterior permite una mayor escalabilidad y simplicidad para su implementación, sin embargo, si el sistema no está preparado y la cantidad de estados crece en gran medida, se puede generar un caos difícil de mantener. Otro problema clásico de este modelo es la oscilación de estados, lo que ocasiona una transición frecuente entre distintos estados, provocando movimientos o acciones repetitivas que fácilmente rompen con la ilusión de inteligencia.

Existen diferentes variaciones de las máquinas de estados cada una con sus ventajas y desventajas que permiten agregar mayor complejidad a las conductas adaptándose de mejor manera a distintas arquitecturas. Algunas de estas variaciones son:

- *Máquinas de Estados Jerárquicas.* Cada estado de una FSM puede contener otra FSM con distintos estado que definan la conducta del estado mayor.
- *Máquinas de Estados Basadas en Mensajes y Eventos.* Evitando la sobrecarga de buscar las condiciones de transición en cada actualización, este tipo de FSM recibe mensajes de otros objetos indicándole cuando debe hacer una transición a otro estado.
- *Máquinas de Estados Difusos o con Transiciones difusas.* Estas máquinas extraen conceptos de la lógica difusa en dos aspectos: con la posibilidad de estar en más de un estado simultáneamente o utilizando funciones difusas para realizar la transición entre estados.
- *Máquinas de Estado basados en Pilas.* Estas máquinas establecen su estado actual a través de una estructura tipo pila, es decir cada nuevo estado es introducido a la pila en cada transición y es extraída una vez que la lógica del estado fue extraído por completo.
- *Múltiples Máquinas de Estado Concurrentes.* Con un inherente problema de coordinación existen dos categorías: para controlar múltiples agentes o para controlar distintos aspectos de un mismo agente.

3.4.3. Sistemas Basados en Mensajes

Uno de los modelos más utilizados, tanto como las máquinas de estados son los *sistemas basados en mensajes*, los cuales aunque no son una estructura para la toma de decisiones son extremadamente útiles para los sistemas de IA. Sus beneficios surgen con el uso de envío de mensajes como método de comunicación entre los distintos componentes que conforman el sistema.

Debido a la sobrecarga que se genera cuando un grupo de agentes busca cambios en el entorno en cada actualización, un sistema basado en mensajes es el elemento más adecuado, ya que facilita la interacción no solo con los diferentes componentes del juego, sino también con los agentes. Lo anterior se logra enviando información relevante solo cuando ésta es necesaria. De esta forma un agente puede enviar un mensaje a otro, indicando que esta atacándolo con una cantidad definida de fuerza. El agente que representa su enemigo, recibe el mensaje procesando la lógica que disminuye su energía en forma racional a la información contenida en dicho mensaje. Lo que a su vez genera un

nuevo grupo de mensajes, uno de los cuales puede indicarle a un modulo que debe iniciar la animación correspondiente, mientras que otro puede informar a un sistema de IA que considere la situación actual para tomar una acción que persiga sus metas internas, dentro de las cuales puede encontrarse la supervivencia.

3.4.4. Sistemas de Información Basados en Posición

Los *sistemas de información basados en posición*, más que una técnica para la toma de decisión es un conjunto de herramientas que provee información adicional al motor de IA. Este conjunto de datos es almacenado de forma centralizada e incrementa la eficiencia del sistema permitiendo que cada agente obtenga información pública, principalmente del mundo que lo rodea.

La meta primaria es contener información sobre las posiciones dentro del mundo virtual. El término *mapa de influencia* se refiere al uso de un arreglo de datos, donde cada elemento representa información específica de una posición en el entorno. Así el módulo más beneficiado es aquel que controla el movimiento autónomo de los agentes pues podrá acceder a información relevante de cada posición sobre la que cada agente potencialmente pueda ubicarse. De esta forma se calculan los recursos necesarios y probabilidades de riesgo potenciales, permitiéndole calcular una ruta más adecuada.

Otro esquema relacionado con los sistemas de información consiste en colocar no sólo datos, sino también fragmentos de lógica distribuidos en distintos objetos presentes en el entorno. Con esta técnica un agente puede acercarse a un objeto y éste le proporcionará información sobre su utilidad, características, además de un conjunto de acciones que pueden indicarle al agente la forma en como debe interactuar con el objeto.

3.4.5. Arquitecturas basadas en metas

En una *arquitectura basada en metas* durante cada actualización del sistema los agentes examinan el estado del entorno y seleccionan de un grupo predefinido una meta de alto nivel o una estrategia. Aquella que satisfaga su deseo más fuerte en el momento, el agente tratará de seguirla descomponiéndola en submetas, cumpliendo cada una de éstas a la vez. Y continuará haciendo esto hasta que la meta sea alcanzada o hasta que el estado del juego indique que la meta no puede ser cumplida, o finalmente se requiera de un cambio de estrategia.

Debido a que cada submeta debe ser terminada en turno, estas arquitecturas establecen una estructura jerárquica que permite seguir la secuencia de acciones, por ejemplo, supongamos que un personaje necesita una espada para

matar un dragón. La estructura que demuestra la secuencia de acciones que realiza puede verse a continuación:

- Comprar Espada
 - Conseguir Oro
 - Planear camino hacia la mina
 - Seguir el camino
 - Levantar pepitas de oro
 - Ir a la Herrería
- Matar Dragón

Como se muestra la meta de *Comprar Espada* es dividida en dos submetas: *Conseguir Oro* e *Ir a la Herrería*, la primera segmentada en acciones llamadas *atómicas*, es decir, acciones que el agente ejecuta como una unidad sin poder dividirla en más subacciones; las que secuencialmente ejecutará una a una, hasta conseguir la meta superior o algún evento en el entorno lo obligue a reconsiderar sus metas; como podría ser que en el transcurso del camino se encuentre con un enemigo que lo fuerce a huir.

3.4.6. Otras técnicas

En busca de la obtención de comportamientos más realistas y retadores, desarrolladores experimentan con diferentes técnicas en sus videojuegos. Algunas de éstas son los algoritmos genéticos, las redes neuronales y los sistemas expertos.

Un *algoritmo genético* trata de simular el proceso natural de la evolución [KURA07] haciendo una abstracción de las características que representan a un individuo. Lo anterior a través de un proceso iterativo que evoluciona a un individuo, que es capaz de resolver el problema en cuestión. En [HERE08] se presenta un análisis de cómo pueden usarse estos conceptos para manipular el comportamiento de una sociedad de agentes dentro de un entorno virtual.

Las *redes neuronales* son otro ejemplo de un modelo basado en procesos biológicos, simulando la interacción de las neuronas que forman el cerebro de los seres vivos. Aunque su desempeño es claramente muy lejano al del cerebro humano, el beneficio real de una red neuronal se encuentra en el reconocimiento de patrones y predicción de eventos. Llevando esto al campo de los videojuegos, un jugador, en un juego de estrategia, puede organizar un grupo de agentes preparándose para atacar al enemigo, entonces el motor de IA puede reconocer estos movimientos como un patrón de ataque que le da la clave para iniciar una acción de defensa, lo que podría ayudar a predecir un momento de vulnerabilidad del jugador que puede ser aprovechado para derrotarlo.

Una red neuronal debe ser sometida a un proceso de entrenamiento que le permita reconfigurarse internamente para que finalmente pueda reconocer un patrón definido. Esta característica puede utilizarse como un proceso de aprendizaje que le da a los agentes una propiedad de adaptabilidad acorde a las distintas situaciones que se presenten en el juego.

Un sistema experto integra bases de datos, memorias, mecanismos de razonamiento, algoritmos, heurísticas, para adquirir, generar y almacenar conocimientos inicialmente adquiridos a través de varios expertos humanos dentro de un dominio específico [MARJ08]. En el caso de los videojuegos dicho dominio se encuentra ligado al tipo y trama del juego, donde es muy común que las empresas desarrolladoras contraten expertos en distintas áreas. Un videojuego inspirado en los conflictos de la segunda guerra mundial podría utilizar los conocimientos de historiadores o militares, y alimentar un sistema experto que pueda tomar acciones que controlen a los agentes para que estos actúen de una manera razonable según el contexto. En [CHRM02] se presenta una aproximación de cómo este tipo de sistemas pueden ser implementados en videojuegos.

Debido a la diversidad de problemas que pueden presentarse en los videojuegos, las técnicas usadas varían enormemente. Sistemas probabilísticos y estadísticos [TOZP02] pueden ser utilizados tanto en el razonamiento para seleccionar la mejor estrategia de juego como para decidir cuál será el mejor camino para llegar a un punto definido, sin embargo, cada problema cuenta con un conjunto específico de características y restricciones lo que deja en manos de los desarrolladores el análisis para encontrar la técnica más adecuada que les proporcione los resultados más adecuados a sus objetivos.

3.5. ¿Qué es un videojuego afectivo?

3.5.1. Definición

De acuerdo a [GILK05] en un *videojuego afectivo* tanto el jugador como el videojuego mismo tienen que responder a las señales afectivas de el otro. Para los jugadores esto siempre ha sido así, pues como seres emocionales responden a la estimulación del juego (disfrutando de la experiencia) tanto como a cualquier otro elemento afectivo (como la historia o los personajes). El sistema por otra parte, la única forma de obtener información del jugador es por las señales usuales, es decir usando el ratón, teclado o joystick para indicar instrucciones como: mover izquierda, mover derecha, etc.

En un intento por proveer al sistema de información relativa al estado afectivo del jugador se introdujeron técnicas de *bioalimentación* que consisten en tomar lecturas biológicas del usuario como su ritmo cardíaco o la conductividad

eléctrica de la piel. Los videojuegos que utilizan este sistema controlan aspectos del juego usando estas señales, así un jugador puede moderar la velocidad de un personaje entrando en un estado de tranquilidad disminuyendo la frecuencia de sus latidos cardiacos, sin embargo, rápidamente los jugadores aprenden a controlar sus respuestas fisiológicas lo que transforma a la *bioretroalimentación* simplemente en otro tipo de dispositivo de entrada.

Para que realmente podamos considerar a un videojuego afectivo, éste debe leer distintas señales del jugador de forma impredecible con el objetivo de ajustar ciertas características o parámetros del juego en progreso. Por ejemplo, algunos juegos podrían leer algunos signos del jugador, percibiendo su ritmo cardiaco. Si el juego detecta un estado de tranquilidad podría disminuir la luz de entorno para crear una atmosfera de ansiedad que en otras circunstancias no estaría presente.

Uno de los problemas que surgen con los videojuegos afectivos en la forma en que los hemos descrito es que no es usual contar con una interfaz que permita tomar lecturas de las distintas variables biológicas, esto requeriría de hardware especializado que no se encuentra en la mayoría de los juegos comerciales. Sin embargo, una forma de lidiar con esto es analizar las distintas acciones y comportamientos que el jugador realiza dentro del juego introduciendo información de la manera tradicional. Ciertos patrones en la conducta pueden indicar su estado emocional, por otro lado, el mismo sistema puede guiar al usuario por situaciones específicas, prediciendo hasta cierto punto un estado afectivo esperado.

3.5.2. Clasificación

Un videojuego afectivo puede clasificarse si tomamos en cuenta el objetivo principal por el cual se busca la obtención de algún estado emocional del jugador.

3.5.2.1. Asísteme

Los juegos que caen en la categoría de *Asísteme* tienen como objetivo ayudar al usuario en ciertos aspectos del juego, si por ejemplo, el jugador no puede resolver un acertijo o no puede encontrar el camino a seguir (algo muy común en juegos actuales por la extensión de los ambientes virtuales) entonces el sistema puede proporcionar algunas pistas o incluso mostrar el camino que se debe seguir.

El principal problema es detectar estas circunstancias específicas de cada juego, apenas poco después de que un sentimiento como la frustración o desesperación por no poder proseguir se presente, de lo contrario, esto podría

ser causa del total abandono del juego. Sin embargo, mostrar información a un jugador que no la necesita puede provocar irritación o incluso aburrimiento ya que el juego al mostrarle lo que debe hacer puede perder su factor de reto y diversión.

Esta idea no es nada nueva, juegos como *Teenage Mutant Ninja Turtles* (1989) presentaba una señal a los jugadores (Figura 3.2) indicándoles la dirección en la que debían seguir avanzando, aún los juegos más recientes presentan estas características pues debido a la complejidad que ahora subyace en ellos este tipo de ayuda es casi una norma a seguir en la industria.



Fig. 3.2 Señal de ayuda presentada en el juego T.M.N.T.

3.5.2.2. Régame

En orden de que un juego sirva a sus propósitos, una de sus cualidades principales es proveer al jugador de un reto que pueda disfrutar, no sólo al enfrentarlo sino también al superarlo, sin embargo, la naturaleza de los jugadores establece que cada uno posee diferentes habilidades, ya sea por ser un jugador casual que incluso puede no dominar los aspectos básicos o un jugador experimentado (*hardcore gamer* es un término usual para referirse a estos jugadores) que no solo ha asimilado las nociones básicas sino que incluso es capaz de realizar técnicas avanzadas con relativa facilidad.

La manera en que la industria ha lidiado con la diversidad de jugadores fue incorporando distintos niveles de dificultad en los juegos, usualmente tres: fácil, normal y difícil. Aunque esta solución ha sido realmente exitosa, aún la diversidad de jugadores y la razón que motiva a cada uno de ellos, puede provocar un conjunto de emociones negativas. La presencia de la frustración en un jugador con poca experiencia puede provocar la deserción del juego, de igual manera un *hardcore gamer* puede caer rápidamente en un estado de

aburrimiento si el juego, aún en la dificultad más alta, ya no le presenta ningún reto.

Videojuegos que caen en la categoría de *Rétame* (*Challenge me*) pretenden leer señales del jugador en la búsqueda de emociones como la frustración y el aburrimiento con el objetivo de adaptar su estado interno de forma que pueda ajustar el nivel de dificultad dinámicamente conforme progresa el juego proporcionando así una mejor experiencia a cada tipo de jugador.

3.5.2.3. Emocióname

Los videojuegos han provocado sinceros estados afectivos por medio de sus historias y situaciones, sin embargo, no todos tienen el mismo éxito, aún cuando su contenido está bien diseñado los jugadores frecuentemente caen en experiencias emocionales no esperadas. Con la habilidad de medir el estado afectivo del jugador en un momento dado, el juego nuevamente puede modificar su contenido con el objetivo de evocar la emoción esperada. Esta categoría es llamada *emocióname* (*emote me*) debido a que el objetivo es asegurar que el jugador sea llevado por un grupo de emociones previamente establecidas, de forma que su inmersión en el juego sea aún mayor.

Los juegos con una historia compleja rodeada de una abundante cantidad de personajes son más aptos para esta categoría. Debido a su interacción con el jugador que se ve obligado a buscar información y tomar decisiones relevantes para seguir avanzando en el juego. Debido a las limitantes actuales para obtener correctamente el estado afectivo del jugador, cada juego toma distintas aproximaciones para suponer o predecir las emociones activas, juegos como *Fable 2* (2008) expone al jugador a la idea de escoger su propio camino con el cual formará la personalidad de su avatar lo que en conjunto con un sistema de expresión de emociones limita el dominio de los estados afectivos que pueden presentarse. De esta forma el estado del juego puede alterarse para responder de una manera apropiada.

3.6. Técnicas para representar emociones en videojuegos

3.6.1. Diseño

Una de las primeras formas usadas para incorporar emociones fue a través del diseño mismo del juego. Debido a sus características para contar historias, la narrativa [PEIF05] ha sido una herramienta vital para generar elementos indispensables que atrapen al jugador ya sea por los personajes, la presentación

de todo tipo de conflictos, historias fantásticas o de la vida real, etc. La interacción también hace que los videojuegos sean diferentes de las películas. Lo que hace necesario tomar en consideración estas diferencias al diseñar el videojuego con el fin de incorporar distintos estados afectivos [FRED03].

Otras aproximaciones de diseño se basan en teorías psicológicas y en el entendimiento de las distintas variables del juego [COOD07] [BURS08] para establecer el origen de las emociones a fin de incorporar estos elementos. Una forma es estableciendo una historia con similitudes a la realidad o al entorno del jugador al que va dirigido; personajes con problemas reales o al menos conflictos sobre los que se pueda hacer una similitud a la vida real, búsqueda de recuerdos que por si mismos tengan una carga emocional, conflicto con normas morales, etc. Así cada elemento incrementa el potencial de inmersión al juego, lo que ayuda al surgimiento de emociones, y a su vez aumenta la sensación de inmersión lo que indudablemente genera emociones más intensas.

3.6.2. Simulación de Emociones usando Máquinas de Estados Finitos

En [CHAA03] se presenta una implementación de emociones virtuales por medio de máquinas de estados finitos. En esta simulación se afirma que se incrementa la versatilidad del comportamiento de un NPC. Inicialmente se supone al sistema con la capacidad de expresar emociones de alguna manera y el motor de IA es solo responsable por su manejo. Es entonces necesaria una interfaz que permita comunicar las emociones al sistema para que éstas sean interpretadas, por ejemplo por el motor de animación que ejecutará la animación más adecuada. Esta interfaz se implementa por medio de una función con la cual se puede obtener información de la emoción que sea especificada:

```
float GetValue(const string& emotion);
```

Dada la interfaz, esta propuesta establece que el camino más adecuado para la expresión de las emociones es a través de las acciones debido a que para un observador externo estas son aparentemente más obvias, así al ejecutar la acción de caminar, ésta puede ser triste o alegre y en cualquier caso se podrá observar una diferencia. De igual manera los sentidos son afectados por un estado emocional donde se identifica que el más afectado es el sentido de la vista, el cual también es el más utilizado por los agentes virtuales.

Con estas suposiciones, el siguiente paso para modelar las emociones es definir un conjunto finito de ellas para representarlas. Estarán definidas de forma binaria, es decir, una emoción podrá encontrarse en un estado activo o inactivo. Posteriormente, se dictamina la lista de estímulos externos o internos que provocan cambios en cada una de las emociones, así como las respuestas

sobre el comportamiento que producen. Un ejemplo de la definición de uno de estos estados propuestos por esta implementación es el siguiente:

```
<state name="surprise">
  <transition input="door_open" target="anticipation" />
  <transition input="explosion" target="fear" />
  <output
    precision="0.4"
    delay="0.1"
  />
</state>
```

Descrita en formato XML, este estado representa la emoción de *sorpresa* (surprise) y define dos transiciones hacia otros estados con la presencia de una entrada única y específica. La salida esta expresada como un conjunto de valores específicos para cada uno de los parámetros usados para controlar los comportamientos, en este caso, *delay* o *retraso*, puede referirse al tiempo de reacción ante un acontecimiento que el agente puede tener en su comportamiento.

La generación de una capa de emociones separada de los comportamientos definidos para cada agente permite una flexibilidad que puede ser aprovechada por otros módulos, aunque una implementación por medio de máquinas de estado sufre de las mismas limitaciones que éstas, como la imposibilidad de tener dos emociones simultaneas o una transición rápida, repetida y constante entre emociones. La representación de emociones binarias no da paso a los distintos niveles que naturalmente se presentan, las emociones implementadas de esta forma son bastantes predecibles y permanecen inalterables durante todo el juego, además incluir una sola emoción adicional involucra un desarrollo complejo que no es fácil alterar ni mantener. Algunos de los inconvenientes presentados por esta técnica pueden resolverse usando distintos tipos de máquinas de estados, como máquinas de estado no deterministas, modelos probabilísticos o lógica difusa, sin embargo, éstas solo agregan más complejidad al sistema.

3.6.3. Variables, objetos y los Sims

*The Sims*⁵ es un juego de estrategia y simulación de vida que le permite al usuario tener el control sobre distintos personajes con características muy similares a las humanas con el objetivo de guiar sus vidas en distintas etapas y

⁵ La primera versión de este juego salió a la venta en el año 2000, desde entonces ha sido tal su popularidad que numerosas expansiones, han incrementado la vida de este juego que para el año 2009 se espera la tercera entrega de esta serie que incluye una mejora sustancial en la personalidad y deseos de los personajes. Sin embargo, al hablar de los Sims nos referiremos únicamente a la primera versión del juego.

aspectos como satisfacción de sus necesidades fisiológicas, el cumplimiento de metas, relaciones sociales, etc.

Como una similitud humana los Sims tienen un conjunto de sensaciones, las cuales son modeladas usando curvas de respuesta [KENF02] [CHAA07₂] medidas en un rango de [-100, 100]. Estas sensaciones son clasificadas en dos grupos:

- **Físicas:** Hambre, comodidad, higiene y Vejiga.
- **Emocionales:** Energía, Diversión, Social y Habitación.

El entorno de estos agentes esta rodeado por una cantidad significativa de objetos que por si solos, exponen su utilidad y manipulación de forma que indique con un valor numérico en que grado afecta a cada una de las sensaciones descritas. Por ejemplo, un objeto que representa a un refrigerador tiene una propiedad que indica que afecta de manera positiva con un valor de 20 unidades a la sensación de hambre.

Aunque varias de estas sensaciones no parecen tener relación con las emociones en el sentido en que se han estado manejando en este proyecto, la satisfacción o insatisfacción de cada una de ellas puede generar un comportamiento en los personajes que puede interpretarse como un estado afectivo. De hecho, el conjunto completo de sensaciones son mapeadas a un estado de humor o estado de felicidad (Figura 3.3) usando las curvas de respuesta lo que produce en los agentes un cambio de comportamiento acorde a estos valores. Así un personaje con un rango muy bajo en sus niveles de socialización, constantemente expresa su humor con espontaneas muestras de emociones como la tristeza reproduciendo una animación que asemeja a un llanto. La forma en como esto afecta un modelo de toma de decisiones es que el agente puede rehusarse a ejecutar ciertas acciones, en el caso de que previamente alguna de sus sensaciones valuada en rangos negativos no sean satisfechas en alguna proporción, es decir, que su valor tome un nivel positivo. De esta forma las curvas de respuesta permiten identificar cual de estas necesidades o sensaciones es más importante para satisfacer.

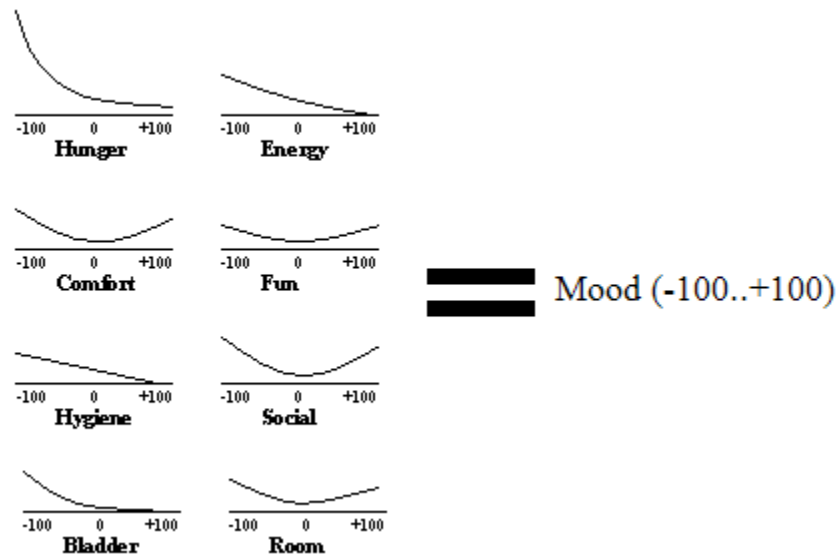


Fig. 3.3 Modelo emotivo del juego The Sims

Desafortunadamente, el estilo de este juego no permite que los personajes basen todas sus acciones en un modelo emotivo, ya que no es su objetivo. Aunque las emociones son obligatorias si consideramos la creación de personajes con similitudes humanas, esta aproximación aún deja de lado algunos aspectos como la distorsión de las diferentes acciones según la emoción activa al momento o la alteración de los sentidos.

3.7. Nuestro caso de estudio

En este capítulo hemos visto cómo los videojuegos han evolucionado en sus aspectos afectivos y como las distintas aproximaciones han mostrado ventajas y desventajas. Nuestra propuesta hace a un lado aspectos como la historia o la expresión de emociones por medio de animaciones o modulación de voz, para concentrarse en un aspecto que a nuestro parecer ha sido descuidado en su desarrollo. La influencia de las emociones en el modelo de razonamiento y toma de decisiones en los NPC de un videojuego, creemos es un elemento más que puede incrementar la inmersión hacia los juegos proporcionando personajes que aparenten estar *vivos* controlando sus comportamientos de una manera más humana y no tan lógica o determinista como normalmente se presenta.

Como plataforma de experimentación, presentaremos un videojuego desarrollado con el objetivo de mostrar los conceptos expuestos. Aunque este juego no califica totalmente como un videojuego afectivo pues no percibe las emociones del jugador. Intenta perseguir de una manera más general una de sus metas al alterar el estado del jugador a través de los sucesos acontecidos en el juego, causados principalmente por este último. Lo anterior con el objetivo

de influenciarlo con ciertas emociones a través del realismo de los comportamientos de los personajes.

El estilo y la dinámica de nuestro juego será presentado en posteriores capítulos donde se hará mención sobre los distintos componentes que lo conforman. Cada uno de ellos ha sido desarrollado tomando en consideración el estado actual de los videojuegos. El propósito es desarrollar una propuesta de arquitectura que nos permita el uso de estados afectivos sin comprometer de manera importante el diseño actual de estos sistemas.

La arquitectura propuesta, descrita en el siguiente capítulo, se basa en el origen cognitivo de las emociones propuesto por la *teoría OCC*. Esta abstracción nos permitirá crear una capa separada que facilite la administración de las emociones. La cuál proporcionará información a otros módulos con el fin de crear comportamientos emergentes, o en algunos agregar nuevas tareas con una característica afectiva más realista. El módulo de razonamiento será el mayor beneficiado con esta propuesta de modelo.



Capítulo 4

Arquitectura Emotiva*

4.1. Necesidad de una Arquitectura Emotiva en Videojuegos

En los capítulos anteriores se han identificado varios de los motivos que han inspirado este trabajo. Durante nuestra investigación hemos encontrado diversos beneficios en el uso de estados afectivos en los agentes para realizar distintas tareas. Esto nos ha sugerido la necesidad de incorporar estas ideas en un ambiente virtual como lo es un videojuego que necesita encontrar nuevos métodos para incrementar la credibilidad y la inmersión de los jugadores.

La programación de videojuegos tiene un nivel de desarrollo bastante elevado donde existen muchas técnicas que han mostrado su eficacia durante años y aunque siempre surgen algoritmos nuevos, en la mayoría de los casos éstos no sustituyen por completo a los más utilizados. Tomando esto en consideración, planteamos una arquitectura más apegada a los sistemas que comúnmente encontramos en los videojuegos con la finalidad de que nuestras ideas del uso de emociones sea un complemento más que un nuevo esquema de desarrollo.

Así partimos de la arquitectura básica de un sistema de este tipo. Rabin [RABS00], menciona que durante la ejecución de un videojuego los componentes de IA entre otros, necesitan que sus objetos (unidades individuales que lo conforman) se actualicen cada cierto tiempo, además, estos objetos deben comunicarse unos con otros. Para lograrlo cada objeto podría estar activamente percibiendo su entorno lo que se conoce por *sondeo (polling)*

* Los contenidos presentados en este capítulo amplían las ideas aportadas en la publicación referenciada como [ACED08].

o podría estar inactivo hasta recibir una señal externa de otro objeto, es decir, un sistema *manejado por eventos* o *event driven*. Dada la gran cantidad de objetos que se encuentran en los videojuegos actuales la solución más utilizada es diseñar una arquitectura manejada por eventos, la cual además nos permite hacer una interesante analogía con los eventos generados dentro del juego y los acontecimientos descritos por la teoría OCC que puede percibir un agente o en nuestro caso un NPC.

4.2. Descripción General

Antes de introducir nuestra propuesta es necesario definir ciertos conceptos que nos ayudaran a entender su funcionamiento, éstos además representan algunos de los módulos propuestos que incluyen las ideas descritas en capítulos anteriores:

- *Entorno*. Al igual que en un sistema multiagentes el entorno es el mundo o el ambiente, donde se alojan los objetos y agentes. Los agentes por medio de sus acciones alteran el entorno al perseguir sus objetivos [LAUA03].
- *Evento*. Señales que utilizan los objetos dentro del sistema para comunicarse unos con otros, llevando toda la información necesaria para su procesamiento. Este es un componente básico e indispensable sobre el cual funcionan los sistemas basados en mensajes.
- *Acontecimiento*. Con el mismo sentido que lo describe la teoría OCC, es una creencia acerca de lo que sucede en el entorno. Esto incluye también el resultado de las acciones realizadas por agentes [LAUA05].
- *Generador de Acontecimientos (GA)*. Siguiendo el esquema de sondeo este objeto percibe todo lo que acontece dentro del entorno. Su objetivo es generar eventos los cuales encapsulan una abstracción con información de los acontecimientos de relevancia para los agentes que se espera experimenten emociones.
- *Habilidades*. Dentro del contexto del videojuego las habilidades están representadas por las capacidades del personaje con efecto directo en su entorno. Por ejemplo: caminar, disparar, recoger objetos, etc.
- *Comportamientos*. Es un conjunto de habilidades bien definidas que se completan en secuencia, por ejemplo: atacar que puede incluir habilidades como caminar, seleccionar un objetivo y disparar. Generalmente, cada comportamiento se genera con el objetivo de cumplir una meta específica y debido a los cambios frecuentes del entorno es común que un comportamiento no llegue a su término.

- *Módulo Afectivo Individual (MAI)*. Este módulo define una instancia única que representa las emociones así como las intensidades que manifiesta un individuo (en este caso los agentes). Además cuenta con una representación de las metas, normas y actitudes. Se encarga de valorar: 1) el impacto de los acontecimientos en las metas, y 2) las acciones de los agentes con base en las normas. Lo anterior con el fin de diagnosticar las distintas emociones del agente y reajustar sus metas.
- *Sistema Cognitivo (SC)*. Se encarga de tomar la información relevante y necesaria para producir un comportamiento en el agente. Este componente es común en cualquier implementación de IA contenida en un videojuego. Sus implementaciones son variadas como se describió en el capítulo anterior, y el único cambio respecto a nuestro diseño es que puede obtener información relevante a las emociones. El comportamiento real de los agentes será determinado por la implementación del SC.

La Figura 4.1 representa el concepto general de nuestra propuesta.

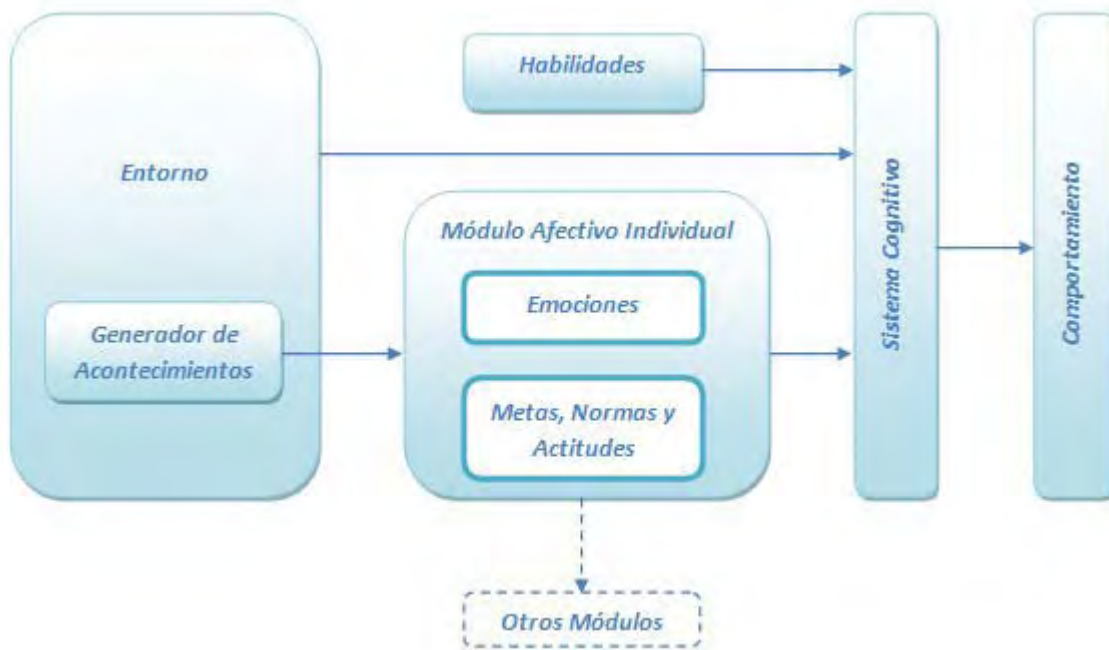


Fig. 4.1 Arquitectura Emotiva

En todo ambiente virtual, que en el caso de estudio se refiere a un videojuego, existe una representación lógica del mundo donde se encuentran todos los objetos y agentes junto con sus propiedades, éste entorno se encuentra en continuo cambio conforme el juego se va desarrollando, el *Generador de Acontecimientos* monitoriza constantemente el entorno en busca de cambios

que puedan interpretarse como acontecimientos relevantes para los agentes, en este caso, dichos acontecimientos deben ser definidos con base en las reglas y características del juego. Este módulo es la primera idea relacionada con el marco teórico, pues identifica y agrupa el conjunto de acontecimientos que según el modelo OCC serán sometidos al proceso de valoración por parte de cada individuo. El *Modulo Afectivo Individual* recibe los eventos generados por el GA para realizar el proceso de valoración de las emociones con base en las metas, normas y actitudes. Este proceso genera un valor de intensidad por cada emoción considerada lo que según el modelo OCC será necesario para determinar si cada emoción en particular se encuentra activa o no [ACED08]. El siguiente paso en el sistema muestra que el *Sistema Cognitivo* recibe información del bloque de *Habilidades*, con lo que puede determinar lo que el agente puede o no hacer para modificar su entorno. En este caso, es necesario hacer hincapié en que el SC tiene acceso a información del entorno (evidencias) que le permite ver si es posible realizar cierta habilidad en las condiciones actuales, por último, el SC recibe información sobre las *emociones* presentes en el MAI, así como las *metas del agente* con lo que ahora cuenta con todo lo necesario para generar un comportamiento adecuado. Este comportamiento resultante será llevado a cabo hasta su término por el sistema mientras las condiciones del entorno y el MAI no cambien.

Debemos aclarar que si bien el MAI es un bloque único por agente, podría pensarse que el bloque de habilidades y el SC también deberían serlo, sin embargo, si suponemos un juego donde se cuente con un conjunto de agentes de idénticas capacidades, por ejemplo, un grupo de soldados todos armados usando la misma clase de fusil, entonces todos ellos comparten el mismo grupo de habilidades las cuales pueden afectar su entorno. De manera análoga si estos soldados se suponen entrenados de igual manera entonces sabemos que contarán con las mismas habilidades para enfrentar la misma situación. Por lo que el Sistema Cognitivo debe ser idéntico en todos ellos a excepción de las emociones que estén sintiendo.

Finalmente, podemos observar que cualquier otro módulo dentro de la arquitectura del videojuego puede obtener información sobre las emociones presentes en cada agente. Lo anterior con el fin de que algún otro módulo, por ejemplo, de animación pueda identificar una emoción de *ira* afectando el modelo (malla o textura, según se implemente en el juego) correspondiente para que pueda expresar dicha emoción.

4.3. Un Ejemplo Concreto

4.3.1. El Entorno

Para mostrar de manera general el funcionamiento de nuestra propuesta debemos definir el entorno, lo que nos lleva a seleccionar un videojuego.

Inicialmente, escogeremos un juego de deportes, específicamente, básquetbol. Al igual que en la mayoría de estos juegos todos los personajes se encuentran dentro de un entorno finito; en este caso la cancha de juego. Poseen las mismas habilidades básicas que esperaríamos de cualquier jugador en la vida real, es decir: caminar, correr, saltar, botar el balón, hacer un pase, tirar a la canasta, etc.

Para definir el contexto hacemos referencia al videojuego *NBA Street Homecourt* (2007) desarrollado por *Electronic Arts* [EASP08], donde el objetivo del usuario es llevar a un personaje a lo largo de su carrera como jugador amateur, armando su propio equipo, jugando en canchas publicas, participando en torneos locales o defendiendo su cancha de juego frente a otros barrios, hasta algún día convertirse en profesional.

4.3.2. Los Agentes

En este caso, encontramos que el usuario esta acompañado por dos personajes más que forman su equipo. Cada uno de ellos puede experimentar diferentes emociones al igual que otros basquetbolistas que se enfrentan a lo largo del juego, sin embargo, para este ejemplo analizaremos a uno solo de ellos que llamaremos Ben.

Siendo un buen jugador de básquetbol, Ben tiene la meta de convertirse en profesional y dada su personalidad persigue otras metas que lo hacen identificarse como individuo. La Figura 4.2 muestra un diagrama de la estructura virtual de metas que nuestro personaje persigue activamente.

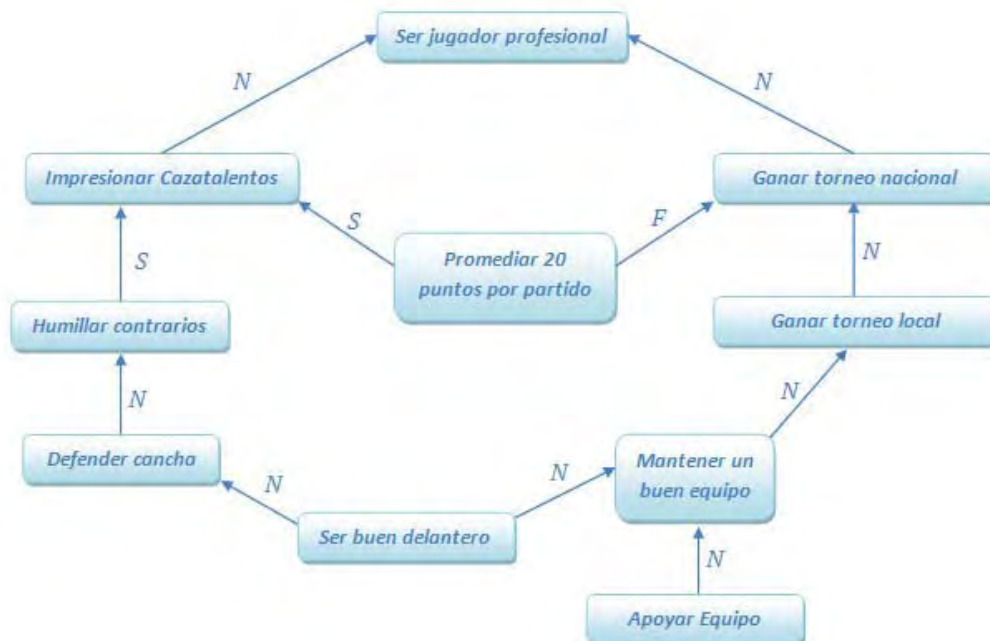


Fig. 4.2 Estructura Virtual de Metas

De acuerdo a la nomenclatura utilizada en [ORTA88], la figura muestra los distintos tipos de enlaces para las metas. Los marcados con una **N** son *necesarios*, lo que desde el punto de vista de Ben significa que para convertirse en profesional debe: 1) ganar el torneo nacional, y 2) impresionar a un cazatalentos. Los enlaces de *suficiencia* (señalados con **S**), le indican a Ben que si desea impresionar a un cazatalentos debe alcanzar al menos una de las submetas establecidas, como es el caso de humillar a los contrarios con un excelente juego. Finalmente, el enlace marcado con una **F** presupone que cuando Ben promedie 20 puntos por partido, le *facilitará* la tarea de cumplir su meta que es ganar el torneo nacional.

4.3.3. La Emoción Activa

Una vez que las metas del personaje están definidas podemos identificar cada una de las emociones que Ben puede manifestar durante el juego. Supongamos inicialmente que en el desarrollo del juego, Ben no recibe ningún pase por parte del jugador, este acontecimiento es detectado por el GA y enviado al MAI correspondiente de cada agente. El MAI que identifica a Ben, analiza la información identificando que el acontecimiento imposibilita su meta de mantener un buen equipo, lo cual le genera una emoción de congoja, como la tristeza con un nivel de intensidad bastante alto pues su meta inmediata no podrá ser satisfecha, lo que además altera la estructura general de sus metas de alguna manera, probablemente eliminando sus enlaces, lo que podríamos interpretar como un deseo momentáneo de no hacer nada. Dada su tristeza el Sistema Cognitivo analiza sus metas actuales, el entorno y junto con las habilidades disponibles genera un comportamiento nuevo que provocará que Ben se mueva más lentamente por la cancha o prefiera evitar el tiro a la canasta, exteriorizando su decepción por el juego. Un nuevo acontecimiento, incluso el paso del tiempo, reorganizará el estado del MAI lo que a su vez producirá un nuevo comportamiento.

En otro escenario podríamos pensar que Ben posee una personalidad agresiva y en ese sentido lejos de estar triste podría experimentar un nivel de ira, lo que alteraría su comportamiento de forma tal que fuera capaz de tomar la iniciativa con alguna acción que le permitiera alcanzar sus metas en la cancha.

Con la anterior reflexión se pretende dejar en claro que con esta forma de modelar la estructura de metas y su relación con un conjunto de emociones, es posible afectar el comportamiento. Tomando en cuenta rasgos característicos de la personalidad que deseamos producir: un traidor, un héroe, un rencoroso, etc.



Capítulo 5

Diseño de una Arquitectura Emotiva Aplicada a un Caso Específico*

5.1. Introducción

Existen muchas áreas donde puede incrementarse el nivel de credibilidad de un videojuego por ejemplo: la animación, la expresividad de los personajes, la interacción con el ambiente, etc. Este trabajo busca obtener credibilidad en el proceso de *planificación de acciones incorporando un modelo de emociones*. Lo cual le permitirá al agente tomar decisiones de manera similar a como un ser humano toma diferentes decisiones, esto es, tomando en consideración las emociones que experimenta.

Al pensar en un modelo de emociones identificamos varios aspectos relacionados con el proceso de toma de decisiones, y que deben tomarse en cuenta para el diseño del sistema:

- *Percepción*: La forma en cómo los agentes perciben el mundo y así mismos es ser influenciado por sus emociones principalmente si éstas son muy intensas.
- *Toma de Decisiones (Planeación)*: Las emociones tienen alta influencia en los planes y acciones que las personas realizan en cada momento. Bajo las mismas circunstancias pero experimentando emociones diferentes *el comportamiento puede ser diametralmente opuesto*. En este

* Los contenidos expuestos en este capítulo amplían las ideas aportadas en la publicación referenciada como [LAUA09]

punto, identificamos dos aspectos, el primero relacionado con las *metas* del agente, es decir, que de acuerdo a las emociones que experimente, los objetivos que persigue podrían tener una jerarquía dinámica; y segundo, cuando el agente se encuentra formulando un plan de acción puede preferir ciertas acciones u objetos para cumplir su objetivo.

- *Ejecución*: La manera en cómo realizamos ciertas acciones. Por ejemplo, la acción de caminar puede verse enriquecida y alterada cuando caminamos tristemente o alegremente, sin embargo, ambos siguen cumpliendo con el fin principal de la acción caminar, que es trasladarse de un punto a otro.

5.2. Entorno

En todo entorno virtual es imprescindible una descripción clara de las características del entorno, no solo para buscar la implementación más adecuada sino también para definir todos los elementos y aspectos con los cuales los agentes podrán interactuar para cumplir su objetivo.

Para éste trabajo, se eligió el contexto de un videojuego bélico. Dos grupos de agentes tratarán de eliminarse unos a otros. Todos los personajes cuentan con el mismo grupo de *acciones básicas*, estas incluyen: *moverse, disparar, golpear, cubrirse y recoger objetos*. Para una descripción más completa de las características del juego propuesto véase el *apéndice A*.

A continuación, describiremos ciertas características del entorno que serán de utilidad para formar un *vector*⁶ representativo del estado del mundo virtual en un instante de tiempo para cada uno de los agentes. Este vector será indispensable para que el agente pueda desarrollar la planificación y así seleccionar el comportamiento más adecuado, con la finalidad de llevar a cabo su objetivo. Es importante notar que este vector formará la representación interna del mundo para un agente en particular y éste dependerá de lo que el agente sea capaz de percibir a través de su sistema sensorial, donde el estado emocional también forma parte de él.

Combate

- *Presencia Enemiga*. Indica la existencia de enemigos en el entorno.
- *Enemigo a la vista*. Ha localizado a por lo menos un enemigo dentro de su rango de visión.
- *Zona peligrosa*. Existe en su cercanía un grupo grande de enemigos.

⁶ En el contexto de este trabajo definimos a un *vector* como un grupo de variables que en conjunto representan las características de un objeto en un instante dado. De esta forma, cada agente puede contener un vector que contenga la interpretación de un objeto en particular y de acuerdo a su percepción.

- *Cubierta cercana.* Existe en su cercanía un punto para cubrirse de ataques enemigos.

Objetos

- *Recarga de Municiones.* Las municiones se encuentran a una distancia cercana y conocida.
- *Recarga de Energía.* Abastecimiento de energía se encuentra a una distancia cercana y conocida.

Compañeros

- *Amigo cerca.* Su compañero se encuentra a una distancia cercana.
- *Solicitud de ayuda.* Su compañero ha enviado una petición de ayuda.
- *Compañero herido.* Su compañero está herido y no puede seguir peleando.
- *Status Equipo.* Todos los miembros del equipo tienen energía y municiones.

Propiocepción

- *Municiones.* Cantidad de municiones con las que cuenta.
- *Energía.* Cantidad de energía que tiene.
- *Cubierto.* Si se encuentra cubierto de ataques enemigos.
- *Enemigo Objetivo.* Enemigo que se ataca, en el estado en curso.
- *Enemigos Eliminados.* Cantidad de enemigos eliminados en el transcurso de la persistencia del agente (tiempo de existencia dentro del sistema).

5.3. Metas

Las Metas u objetivos de acuerdo a [RUSS04] son una descripción del estado del entorno que es deseable para el agente y son perseguidas activamente mediante sus acciones [LAUA00₂], [LAUA03] y [LAUA05]. En nuestro caso de estudio contamos con las siguientes metas para nuestros agentes:

- *Eliminar Enemigos.* Eliminar mediante disparos o golpes a todos los enemigos existentes en el entorno.
 - Vector del entorno: Presencia enemiga = 0
- *Sobrevivir.* Permanecer con cierta cantidad de energía durante la existencia del juego.
 - Vector del entorno: Energía > 0, Zona peligrosa = 0
- *Proteger Equipo.* Ayudar a los compañeros a eliminar enemigos o levantarlos cuando son derribados (esto sucede cuando la variable que representa su salud, llega a cero debido a los ataques recibidos por otros agentes).

- Vector del entorno: Solicitud de ayuda = 0, Compañero Herido = 0, Status Equipo > 0

5.4. Normas

Antes de introducir nuestro modelo emocional, debemos hacer mención de las *normas*. Así como las metas u objetivos que le dan una motivación al agente de realizar cierta acción o comportamiento, una *norma* le permite al agente establecer un contexto sobre: qué acciones, propias y ajenas, son correctas o incorrectas. Por ejemplo, un soldado tiene una norma firmemente establecida que le dice que debe servir a su país. Si observamos las acciones del agente podemos plantear un límite, el cual permite generar un marco que nos indique si el agente esta haciendo lo correcto o no. Las *normas* son importantes en la valoración de ciertas emociones, por lo cual, es necesario que sean consideradas para la arquitectura propuesta. En el caso de estudio se plantea una norma sobre la cual se medirán las acciones de nuestros agentes.

Como lo describen los autores de la teoría OCC para interpretar si una norma es observada o no, es necesario prestar atención a las acciones de los agentes, a través de éstas se pueden interpretar los acontecimientos, que involucran uno o más agentes. Bajo esta perspectiva es posible interpretar el cumplimiento de una norma. De esta forma la norma será formalizada por la presencia o ausencia de un conjunto de acontecimientos relacionados con un agente específico. En el caso de estudio, definimos una sola la norma para nuestros agentes, que se define a continuación:

Los Compañeros deben Ayudarse.

Formada por la ausencia o presencia, según sea el caso, de los siguientes acontecimientos:

- Eliminar Enemigo (jugador). Para que la norma no sea violada, el acontecimiento de *eliminar enemigo* realizada por el jugador debe presentarse con regularidad, es decir por lo menos una vez cada lapso de tiempo t_E .
- Atender Solicitud de Ayuda (jugador). Al igual que la anterior para que la norma sea observada, una solicitud de ayuda debe ser atendida por el jugador antes de que se cumpla un tiempo t_A .
- Ataque Aliado (jugador). En este caso, la presencia del acontecimiento en que el jugador genera un *ataque a su compañero* será causa suficiente para considerar la violación de la norma.

5.5. Acontecimientos

La valoración de los acontecimientos nos permitirá establecer un parámetro sobre el cual se presentaran las emociones. Como se describió en el capítulo anterior, el Generador de Acontecimientos es un módulo en nuestra arquitectura que se encarga de identificar estos acontecimientos y darlos a conocer a los agentes que lo requieran. Esto nos obliga a definir con precisión dentro del sistema, la forma de identificación de estos acontecimientos. En el caso de estudio y con base en las emociones propuestas describimos el significado y características de cada uno de los acontecimientos que se consideran relevantes a los agentes según el contexto del juego.

Dado que los acontecimientos están directamente relacionados con la representación del entorno y los cambios que este sufre con el transcurso del tiempo. Definimos entonces a las variables t_1 y t_2 como el instante de tiempo inicial y final, respectivamente, en los cuales se realizan las mediciones sobre las distintas variables del vector representativo.

Los acontecimientos considerados en el sistema son:

Esperanza

La esperanza puede comprenderse a través del contexto de la situación en que el equipo del usuario ha eliminado tantos enemigos que incrementa el valor del factor de esperanza ante la posibilidad de salir victoriosos debido al liderazgo del jugador. Este acontecimiento se presenta bajo las siguientes condiciones del entorno.

$$EEA(t_1) + EEJ(t_1) - EEA(t_2) + EEJ(t_2) > FE, \text{ y } EEA < EEJ$$

Donde:

FE. Factor de esperanza.

EEA. Enemigos eliminados por el agente en la totalidad del juego.

EEJ. Enemigos eliminados por el jugador en la totalidad del juego.

Enemistad

Suponemos que una situación de enemistad se presenta cuando un agente, a pesar de enviar una solicitud de ayuda, no recibe apoyo de su compañero en un tiempo considerable. Este acontecimiento se define de define con base en las propiedades del entorno. Las variables del entorno que indican: a) una solicitud de ayuda, y b) un compañero esta herido, se presentan simultáneamente durante un tiempo mayor a TE, es decir:

$$SA(t_1) \wedge CH(t_1) = 1, \quad SA(t_2) \wedge CH(t_2) = 1 \text{ y } t_2 - t_1 = TE$$

Donde:

- SA.** Solicitud de ayuda.
- CH.** Compañero herido.
- TE.** Tiempo de Enemistad.

Eliminar Enemigo

Se produce cada vez que el jugador elimina un enemigo.

$$EEJ(t_1) < EEJ(t_2) + 1$$

Donde:

- EEJ.** Enemigos eliminados por el jugador en la totalidad del juego.

Encontrar Enemigo

Se genera cada vez que un enemigo aparece en el campo visual del agente.

$$FOV[PE(t_1)] = 0 \text{ y } FOV[PE(t_2)] = 1$$

Donde:

- FOV.** Función que identifica que un enemigo se encuentra en el campo de visión del agente.
- PE.** Posición del enemigo.

Usar Cubierta

Generado cada vez que un personaje se ubica en la misma posición que una cubierta durante un tiempo determinado.

$$CC(t_1) \wedge CC(t_2) = 1 \text{ y } t_2 - t_1 = TC$$

Donde:

- CC.** Estado en el cual la posición del agente se encuentra a una distancia cercana a la posición de la cubierta.
- TC.** Tiempo mínimo necesario para estar cubierto.

Recoger Energía

Acontecimiento generado al recoger el objeto que regenera la energía de los personajes.

$$EA(t_1) < EA(t_2)$$

Donde:

- EA.** Energía del agente en el tiempo dado.

Recoger Armamento

Generado al recolectar el objeto que recarga las municiones de los personajes.

$$MA(t_1) < MA(t_2)$$

Donde:

MA. Municiones del agente en el tiempo dado.

Encontrar compañero

Acontecimiento producido cuando el usuario y su compañero se encuentran a una distancia cercana durante un tiempo dado.

$$DJA(t_1) < DM \text{ y } DJA(t_2) > DM$$

Donde:

DJA. Distancia calculada entre la posición actual del jugador y el agente que lo acompaña

DM. Distancia mínima establecida.

Eliminar enemigos cercanos a compañero

Se genera cuando se elimina un número dado de enemigos cercanos a la posición del compañero.

$$EEJ(t_1) < EEJ(t_2) + 1 \text{ y } DEEA(t_2) < DM$$

Donde:

EEJ. Enemigos eliminados por el jugador en la totalidad del juego.

DEEA. Distancia entre la posición de los enemigos eliminados y la posición de agente que acompaña al jugador.

DM. Distancia mínima establecida.

Atender Solicitud de Ayuda

Este acontecimiento es producido después de presentarse los acontecimientos de *Encontrar compañero* así como *Eliminar enemigos cercanos a compañero* siempre y cuando la señal de solicitud de ayuda se encuentre activada antes de la presencia de éstos.

Ataque Aliado

Se genera el acontecimiento cuando un proyectil disparado por el usuario colisiona con el compañero del jugador.

$$EA(t_1) > EA(t_2) \text{ y } \text{Colisión}[PDJ(t_2), PA(t_2)] = 1$$

Donde:

EA. Energía del agente que acompaña al jugador.

PDJ. Posición del proyectil disparado por el jugador.

PA. Posición del agente que acompaña al jugador.

5.6. Emociones

Un agente, además de percibir el entorno, tiene también la capacidad de poder medir su propio estado interno (propiocepción), de esta forma es natural extender este concepto implementando un modelo de emociones como una característica más a la estructura interna del agente y por ende al sistema de percepción.

Para lograr esto, identificamos las emociones que diseñamos para nuestro agente utilizando una estructura cognitiva de emociones, es decir, una descripción de su origen, representación e implicaciones, de forma que estas puedan ser integradas en sus comportamientos. Cada individuo (en nuestro caso agente) es único dentro de nuestro entorno, por lo que percibe emociones diferentes y de diferente intensidad. Lo anterior debido a su intransferible mapa emocional (véase capítulo 2, sección 2.1.1 y capítulo 4, sección 4.2 y 4.3.3). Las emociones formarán un vector que enriquecerá el vector del entorno para lograr una representación más completa del mundo y de sí mismo.

Al tratar de incorporar emociones en nuestros sistemas es importante tomar en cuenta los problemas que estos presentan incluso antes de una representación computacional, algunos de éstos se discuten con más detalle en [ORTA88] y [PICR97]. Tomando en consideración lo anterior y enfocándonos en este proyecto, cada emoción que pretenda usarse vendrá acompañada de una descripción y conjunto de características que permitirá al desarrollador exponer una idea clara de cómo la emoción propuesta afecta al agente, y como se esta implementando en el sistema. Es en este punto donde se deja claro durante el proceso de diseño, la personalidad que se desea reproducir al relacionar las emociones con los comportamientos y todo con base en la estructura cognitiva desarrollada (metas, normas y objetos). Estos conceptos son:

Nombre. Un nombre con el cual se identificará a la emoción en el proyecto. En ciertos casos no existe una palabra adecuada en el idioma que permita identificar una emoción específica, sin embargo, es importante contar con una palabra de otro idioma o incluso inventada con la finalidad de identificarla.

Descripción Psicológica. Desde un punto de vista psicológico, existen muchas teorías que pretenden definir y explicar el funcionamiento, e influencia de las emociones sobre el comportamiento humano (véase Capítulo 2). En el caso de estudio sirve como una guía teórica que permite identificar su influencia en el comportamiento y sobre la cual se desarrollará la implementación.

Agente. Dado que una misma emoción puede ser percibida de forma diferente

por cada agente, se definirá la estructura cognitiva de cada uno de ellos, siguiendo los conceptos de la teoría OCC. En otras palabras, este concepto nos indica qué agente en nuestro sistema puede experimentar la emoción descrita, o esta influenciado por la misma.

Representación. Una representación de un tipo determinado (lógica, esquemática, matemática, etc.) que permita su implementación.

Parámetros y Variables. Información sobre todo tipo de parámetro que pueda tener alguna relación directa sobre la emoción. Este aspecto está íntimamente relacionado con el modelo especificado en la descripción psicológica.

Precondiciones. Características específicas del sistema que sean necesarias para dar existencia o alterar la representación de la emoción.

Consecuencias. Repercusión en las distintas áreas del sistema cuando se presenta la emoción.

Ahora contamos con una plantilla de diseño que describirá cada una de las emociones que se utilizarán en nuestro caso de estudio. Ortony, mencionado en [BECC06], nos propone que podemos empezar por darles a los agentes un par de emociones (incluso solo diferenciando entre positivo y negativo), y posteriormente, incrementar su rango emocional a través de distintas técnicas. En nuestro caso, propondremos dar a nuestro agente dos emociones que se describen a continuación.

5.6.1. La Emoción de estar Contento

Con la ayuda de la plantilla de diseño se formaliza el comportamiento del agente relacionado con la primera de las emociones propuestas.

Nombre: CONTENTO

Descripción Psicológica: Definido por la teoría OCC como estar contento por un acontecimiento deseable. Se encuentra clasificado dentro de las emociones de bienestar.

Agente: El NPC que acompaña al jugador.

Representación: Se representa por un nivel de intensidad que toma valores entre 0 y 1. Se considera que la emoción está presente cuando su intensidad rebasa un umbral preestablecido.

Parámetros y Variables: umbral, intensidad, factor de decremento, es decir la velocidad con la cual disminuye la variable de intensidad.

Precondiciones: Incremento de la intensidad por medio de la presencia de los acontecimientos: *Esperanza, Usar cubierta, Eliminar enemigo, Encontrar enemigo, Recoger Armamento y Recoger Energía.* El valor de intensidad debe

estar por encima del umbral.

Consecuencias:

Consecuencias en su conducta

Percepción: sus sentidos son más agudos. Puede ver objetos más distantes.

Toma de decisiones: Reorganiza la jerarquía de metas preestablecida (ver sección 5.9 de este mismo capítulo) tomando en consideración el siguiente orden:

1. Eliminar Enemigos
2. Proteger Equipo
3. Sobrevivir

Ejecución: su velocidad de desplazamiento se incrementa. Su energía se incrementa en un factor constante.

Consecuencias en otras emociones

El factor de decremento de la emoción de *desprecio* se incrementa.

Al observar las consecuencias en la conducta de la emoción de Contenido podría interpretarse que nuestro personaje se convierte en un *suicida* al presentarse esta emoción, debido a la baja prioridad que le da a la meta de sobrevivir. Sin embargo, esto tiene otra interpretación. Cuando se habla de la utilidad que tienen las emociones para los individuos, se argumenta como las personas tratan de tomar decisiones que no tendrán que lamentar más tarde y este proceso de selección tiene una base cognitiva [LOEG03] que implica las diferentes perspectivas desde las que se puede interpretar un suceso y en este caso el papel de las normas es muy importante. Por ejemplo, un individuo a favor de la conservación de las ballenas puede realizar ciertas acciones contra los cazadores, aún en contra de su propia seguridad personal. Estas acciones tienen como objetivo proteger un ideal, que en el caso de verse afectado, podría generar una emoción negativa muy intensa. De ésta forma, un agente que experimenta la emoción de contenido según el diseño expuesto anteriormente, no es un suicida, simplemente podría estar decidiendo que darle prioridad a sus otras metas, en lugar de su meta de sobrevivencia, es lo mejor para el mismo y para sus creencias (metas normas y actitudes).

5.6.2 La Emoción de Desprecio

Nombre: DESPRECIO

Descripción Psicológica: Definido por la teoría OCC como la capacidad de censurar la acción de otra persona. Se considera una emoción de atribución en el grupo de emociones de reproche.

Agente: El NPC que acompaña al jugador.

Representación: se representa por un nivel de intensidad que toma valores entre 0 y 1. Se considera que la emoción esta presente cuando su intensidad rebasa un umbral preestablecido.

Parámetros y Variables: umbral, intensidad, factor de decremento, el agente sobre el cual se está sintiendo desprecio.

Precondiciones: Incremento de la intensidad por medio de la ausencia del acontecimiento *Eliminar Enemigo* así como *Atender solicitud de ayuda* y la presencia del acontecimiento *Ataque Aliado*, todas realizadas por su compañero. El valor de intensidad debe estar por encima del umbral.

Consecuencias:

Consecuencias en su conducta

Percepción: ninguna.

Toma de decisiones: Reorganiza la jerarquía de metas preestablecida (ver sección 5.9 de este mismo capítulo) tomando en cuenta el siguiente orden:

1. Sobrevivir
2. Eliminar Enemigos
3. Proteger Equipo

Ejecución: no envía mensajes de ayuda, permanece a una distancia más lejana del jugador sobre el que recae la emoción.

Consecuencias en otras emociones

El factor de decremento de la emoción *contento* se incrementa.

5.7. Macroestructura de Valoración

Nuestra propuesta maneja un modelo de emociones que requiere de un sistema de valoración (teoría cognitiva de las emociones), es decir, cuando se trata de emociones sabemos que éstas se presentan en cada individuo de manera diferente con una intensidad distinta, lo cual nos indica que existe un proceso de valoración, en el cual se evalúan estímulos y variables involucradas. En consecuencia, debe existir algún medio que nos permita relacionar metas, normas y actitudes (aquellas relacionadas con los objetos) con cada agente que experimente emociones en el sistema desarrollado. Utilizando un diagrama relacional, definimos una estructura llamada *macroestructura de valoración* como apoyo del análisis y diseño de estas complejas relaciones.

Aunque ésta estructura no es nada simple puede simplificarse para nuestros propósitos teniendo una estructura jerárquica, en la cual se visualizan las metas y submetas involucradas. Esta jerarquía es dinámica, ya que considera los eventos del entorno, y éste a su vez es dinámico. Dado que las metas no se persiguen activamente todo el tiempo, sino que cada una tiene un grado de activación, esta estructura nos permitirá a través de la jerarquía visualizar la meta activa en un momento dado y su importancia para el individuo.

Cada nodo en la estructura se relaciona con una meta, los nodos de mayor jerarquía representan las metas más abstractas o generales, mientras que las de más bajo nivel representan las metas más concretas e inmediatas. Los nodos

tienen conexiones de entrada y salida con otros nodos. Las conexiones entrantes representan las metas cuya consecución puede verse afectada por la consecución de las metas de nivel más bajo desde las que viene el enlace.

Para poder interpretar esta estructura debemos exponer los distintos tipos de metas y de enlaces que se pueden presentar y que afectarán su relación con los demás nodos. En esta clasificación se puede encontrar tres tipos de metas:

Metas Activas (A): Estas metas son aquellas que se persiguen activamente con las acciones, las cuales representan las metas más generales, aquellas cosas que las personas les gustaría tener hechas.

Metas de Interés (I): Estas metas aunque existen para el individuo generalmente se relacionan con cosas fuera de su alcance, es decir, son metas que le gustaría que se cumplieran pero no puede hacer nada para conseguirlas. Un ejemplo simple lo encontramos al ver a una persona que tiene como meta ver a sus amigos triunfar en la vida.

Metas de Relleno (R): Estas metas similares a las metas activas tienen una característica única. Una vez que se alcanzan no desaparecen sino que, reaparecen en cierto tiempo u ocasiones. El ejemplo más claro de unas metas de relleno son el comer, dormir, etc.

Los enlaces entre metas pueden clasificarse en los siguientes tipos:

Necesarios (N): Estos enlaces indican una necesidad de que la meta de menor jerarquía debe ser cumplida para poder aspirar al cumplimiento de la meta de mayor jerarquía.

Suficiencia (S): En este caso cuando varias metas están relacionadas para alcanzar una meta mayor, bastará que se cumpla una de ellas.

Facilitan (F): Estos enlaces indican que las metas facilitaran la conclusión de una meta mayor, sin que su cumplimiento sea suficiente ni necesario.

Inhibitorios (I): El cumplimiento de ciertas metas pueden provocar la inhibición de otras, es decir, que dado el contexto o sus consecuencias no permitan el cumplimiento de una meta de mayor jerarquía.

En la Figura 5.1, se presenta nuestra propuesta para el agente que acompañara al jugador durante el juego. Podemos observar las principales metas que el agente persigue señaladas con una **A**, es decir, las metas activas. De igual manera se observa un grupo de metas de relleno (señalado con **R**) que se relacionan de diferentes maneras con las *metas activas*.

Tomemos como ejemplo la meta de *Eliminar Enemigos*. A partir de nuestra macroestructura podemos observar un enlace, marcado con una **N**, que indica la necesidad de que la submeta *Eliminar Enemigo* sea alcanzada antes de poder cumplir nuestra meta de mayor jerarquía. Dado que existe más de un enemigo en el entorno esta submeta debe realizarse con regularidad lo que la convierte en *una meta de relleno*. Siguiendo los enlaces observamos la submeta de *Encontrar Enemigo* la cual es necesaria para cumplir la meta inmediatamente superior. Claramente estos enlaces nos ayudan a identificar una lógica en el proceso pues evidentemente es indispensable encontrar al enemigo antes de tomar alguna acción para eliminarlo. Siguiendo con nuestro ejemplo, observamos la submeta *Usar Cubierta*, la cual se encuentra ligada a otras metas de diferente jerarquía, donde su enlace de tipo **F** nos indica que; si existe la posibilidad de alcanzar ésta meta entonces se nos *facilitará* la conclusión de las metas relacionadas.

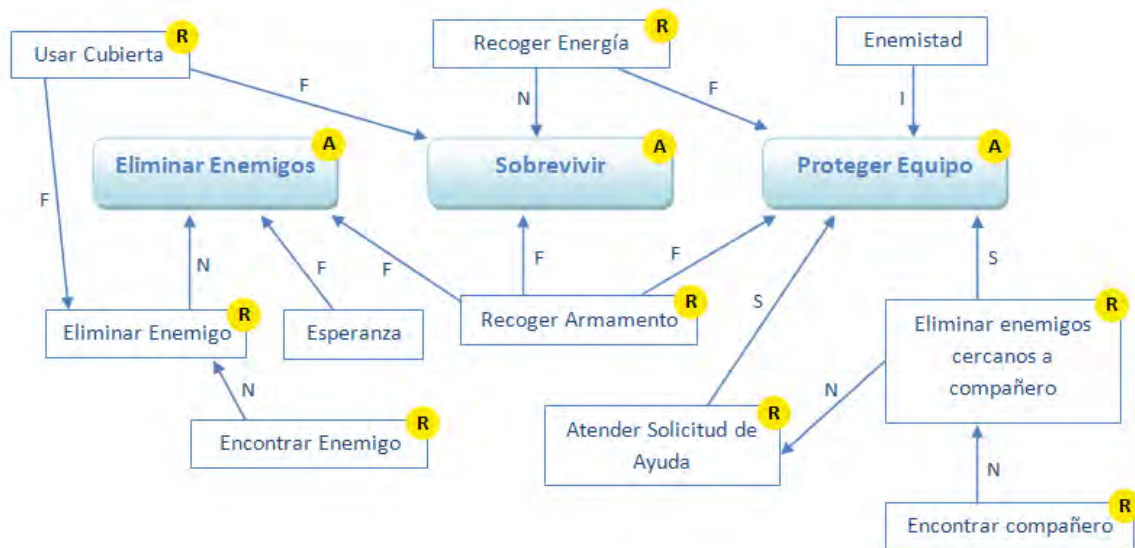


Fig. 5.1 Macroestructura que representa las metas del agente.

Es necesario indicar que cada una de las metas de relleno que se muestran, *tiene un acontecimiento relacionado* (que en este proyecto se identifican con el mismo nombre), es decir, que cada vez que se cumple la meta de relleno *Recoger Armamento*, entonces se está produciendo un acontecimiento que nos informa que el agente recogió del entorno el objeto que le dará más municiones para su arma actual. Este hecho, es de suma importancia pues permite incluir en nuestra macroestructura cualquier acontecimiento que pueda suceder en el entorno. Lo anterior nos da un contexto sobre el cual el agente es capaz de hacer un análisis de los acontecimientos identificando las metas relacionadas y cómo son afectadas. Regresando a la figura, encontramos el acontecimiento *Enemistad*, el cual podemos interpretarlo como la falta de ayuda o comunicación por parte de su compañero, algo que evidentemente no parece ser una meta y que por su enlace del tipo **I** nos señala que la presencia de este acontecimiento inhibe la meta de *Proteger Equipo*, es decir, si pensamos que nuestro

compañero no quiere ayudarnos, o no manda un aviso de que necesita ayuda, entonces será difícil mantener la integridad el equipo.

La macroestructura también debe incluir normas y actitudes, tal como se identifica en la teoría OCC [ORTA88]:

Deben incluirse las normas porque representan las creencias desde el punto de vista de las cuales se hacen las valoraciones de decisiones morales y otras clases, y deben incluirse las actitudes (incluyendo los gustos) porque constituyen la base para la valoración de la capacidad de atraer que es el fundamento del agrado.

Dado este punto, no existe ningún impedimento que nos prohíba incluir las normas, que previamente definimos en la sección 5.4, y ayudarnos de las mismas herramientas para relacionarlo con las metas y/o acontecimientos. De esta forma, nuestra macroestructura queda enriquecida como se muestra en la Figura 5.2.

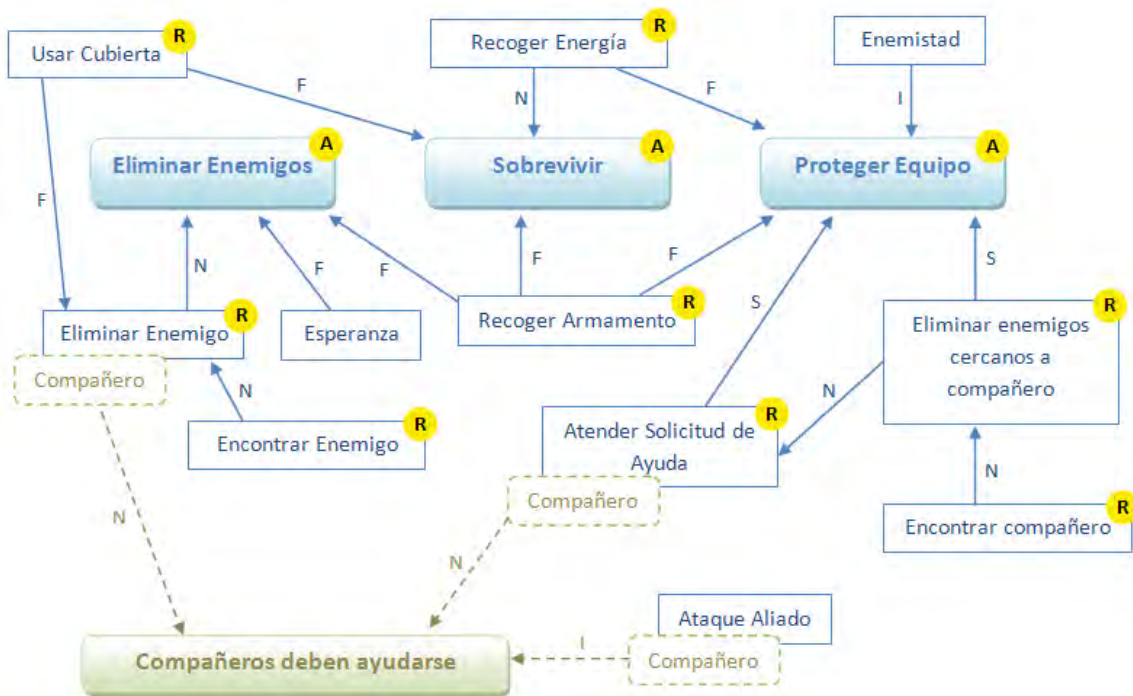


Fig. 5.2. Macroestructura enriquecida por las normas.

Esta vez podemos identificar a las normas y sus enlaces en líneas punteadas. En el caso de estudio la norma especificada es llamada *compañeros deben ayudarse*. Se observa que tiene tres acontecimientos relacionados dos de ellos marcados con una **N**, los que interpretamos como *una necesidad*, es decir, es necesario que estos acontecimientos se presenten para cumplir con la norma, por el otro lado, el acontecimiento *Ataque Aliado* tiene un enlace *inhibitorio* lo que nos dice que éste no debe presentarse si es que queremos que la norma

este presente. Finalmente identificamos un elemento más, encerrado en un rectángulo con líneas punteadas. Este elemento indica al agente responsable de la acción descrita por el acontecimiento relacionado, en el caso del acontecimiento *Ataque Aliado*, este elemento nos indica que el compañero del agente realizó un ataque a su propio compañero lo cual como ya se mencionó implica que se ha violado la norma.

A partir de este punto, contamos con un esquema que nos permite encontrar una relación *entre las metas y normas del agente junto con todos los sucesos presentes en el entorno interpretados como acontecimientos, o en su caso como una dupla acción-agente*. Estas relaciones nos permitirán realizar una valoración según los niveles de las variables de deseabilidad, plausibilidad, etc., de acuerdo a cada emoción. Lo anterior permite obtener un nivel de intensidad con el cual se puede definir la presencia o ausencia de cada emoción en particular. La totalidad del proceso de valoración se lleva a cabo dentro del *Módulo Afectivo Individual* agrupando y exponiendo la información necesaria acerca de las emociones que otros módulos del sistema puedan requerir (ver Figura 4.1. en el Capítulo 4).

5.8. Modelo Cognitivo

De acuerdo a Laureano et al. [LAUA00] y [LAUA01], los modelos cognitivos han sido utilizados con éxito en el análisis: 1) del dominio [LAUA00₂], 2) para modelar conductas reactivas [LAUA00₂] [LAUA01], 3) en la resolución de problemas para representar la forma en que el novato migra hacia la experticia [ARRI01], 4) como parte de un modelo integral que además de incluir el aprendizaje de habilidades cognitivas incluye las afectivas, motivacionales y sociales [CASS99], y 5) en el diseño de currículo [ESTE02]. En Laureano et al. [LAUA00₂] y [LAUA01], se propone un procedimiento para desarrollar un modelo cognitivo. Retomando esta metodología comenzamos desarrollando el Análisis Cognitivo Conductual (ACC) el cual se lleva a cabo con el fin de esclarecer los componentes necesarios para la simulación de comportamientos. Se ha utilizado con éxito en el modelado de comportamientos tanto cognitivos, como físicos. El ACC es un análisis recursivo de comportamientos, que se realiza analizando el proceso psicológico involucrado en la construcción cognitiva del desarrollo de las habilidades; de forma recursiva se divide el comportamiento en subcomportamientos cada vez más específicos, para así conocer con mayor precisión los elementos que lo componen. A estos subcomportamientos se les relacionan los pasos de los procesos mentales que subyacen a cada una. Los procesos mentales están constituidos por estructuras que interactúan, y las estrategias involucradas para lograr cada paso de la tarea.

El Sistema Cognitivo, definido en nuestra arquitectura, fue desarrollado utilizando el procedimiento propuesto por Laureano [LAUA00] y [LAUA01] para formalizar los comportamientos que llevarán a cabo los agentes lo que además

nos será de utilidad para realizar un análisis más profundo en busca de otros elementos que puedan ser relevantes para los estados afectivos. Para mayor información consultar las referencias mencionadas.

Para alcanzar sus metas un agente autónomo cuenta con un grupo de objetivos y *acciones predefinidas* con las cuales puede *alterar su entorno* [LAUA05]. Un conjunto ordenado y coherente de *estas acciones formarán un comportamiento* adecuado para cada meta específica que persiga según la situación existente en el entorno.

Las acciones que nuestros agentes pueden realizar son:

- Esperar (Idle)
- Atacar Lejos (Weapon)
- Atacar Cerca (Melee)
- Atacar Cubierto (Weapon + Cover)
- Pedir Ayuda
- Apoyar Amigo
- Trasladarse a una posición definida
- Recolectar Municiones
- Recolectar Energía
- Buscar (Enemigo, municiones, energía)
- Seleccionar Objetivo

5.8.1. Comportamientos

En nuestra primera aproximación se cuenta con un comportamiento por cada meta que el agente persigue.

Modelo Mental del Comportamiento 1 (meta: Eliminar Enemigos)

Mientras exista presencia enemiga

```

{
  Mientras no haya enemigo a la vista
  {
    Realizar Búsqueda (Enemigo)           Paso 1
  }
  Atacar Enemigo                          Paso 2
}

```

ACC: Comportamiento 1

Pasos	Tipo de Conocimiento	Tipo de Representación
Realizar Búsqueda (paso 1)	Factual, Procedural	Estructuras y Procedimientos
Atacar Enemigo (paso 2)	Estratégico, Procedural	Estructuras y Procedimientos

ACC: Comportamiento 1 (Orientación)

Habilidades	Conocimiento Requerido	Pasos del Modelo Mental
Reconocimiento de Patrones	Tipo de objetos y enemigos	Paso 1 y 2
Estratégico	Situación de combate, cantidad de recursos y generación de rutas	Paso 2
Procedural	Posición de los objetos, rutas	Paso 1

Modelo Mental del Comportamiento 2 (meta: Sobrevivir)

```

Mientras Enemigo no esta a la vista
{
  Si Zona de peligro es alta                                Paso 1
  {
    Alejarse del enemigo
    Pedir Ayuda
  }
  Si Zona de peligro es media                                Paso 2
  {
    Pedir Ayuda
    Si cubierta cercana
    {
      Trasladarse (Cubierta)
    }
    Sino
    {
      Buscar Recursos Necesarios
    }
  }
  Si Zona de peligro es baja                                Paso 3
  {
    Buscar Recursos Necesarios
  }
}

```

ACC: Comportamiento 2

Pasos	Tipo de Conocimiento	Tipo de Representación
Paso 1	Factual, Procedural	Estructuras y Procedimientos
Paso 2	Factual, Estratégico, Procedural	Estructuras y Procedimientos
Paso 3	Factual, Procedural	Estructuras y Procedimientos

ACC: Comportamiento 2 (Orientación)

Habilidades	Conocimiento Requerido	Pasos del Modelo Mental
Procedural	Posición del enemigo, y cómo solicitar ayuda	Paso 1
Estratégico	Cantidad de recursos y generación de rutas	Paso 3
Estratégico y Procedural	Cómo solicitar ayuda, cantidad de recursos, posición de los objetos	Paso 2

Modelo Mental del Comportamiento 3 (meta: Proteger Equipo)

Si existe Solicitud de Ayuda o compañero herido	Paso 1
{ Trasladarse (Amigo)	
}	
Si hay un compañero herido	Paso 2
{ Trasladarse (Amigo)	
Ayudar Amigo	
}	
Si Status Equipo es bajo	Paso 3
{ Buscar Recursos Necesarios	
Si Amigo lejos	
{ Trasladarse (Amigo)	
}	
}	

ACC: Comportamiento 3

Pasos	Tipo de Conocimiento	Tipo de Representación
Paso 1	Factual, Procedural	Estructuras y Procedimientos
Paso 2	Factual, Procedural	Estructuras y Procedimientos
Paso 3	Factual, Procedural	Estructuras y Procedimientos

ACC: Comportamiento 3 (Orientación)

Habilidades	Conocimiento Requerido	Pasos del Modelo Mental
Procedural	Posición del amigo y saber como ayudarlo	Paso 1 y 2
Estratégico	Cantidad de recursos y generación de rutas	Paso 3

Buscar Recursos Necesarios

Si Energía Baja

```
{
    Realizar Búsqueda (Energía)
}
```

Si Municiones Bajas

```
{
    Realizar Búsqueda (Municiones)
}
```

Realizar Búsqueda (objeto)

Si Conoces ubicación de objeto

```
{
    Trasladarse (objeto)
}
```

Sino

```
{
    Buscar (objeto)
}
```

Atacar Enemigo

Seleccionar Objetivo

Mientras Enemigo Objetivo siga vivo y siga en la mira

```
{
    Si existe cubierta cercana y no esta cubierto
    {
        Trasladarse (cubierta)
        Atacar Cubierto
    }
}
```

Sino

```
{
    Si enemigo muy cerca
    {
        Atacar Cerca
    }
}
```

```

    }
    Sino
    {
    Atacar Lejos
    }
  }
}

```

5.8.2. Grafos Genéticos

Los Grafos Genéticos son una herramienta más que enriquecen el Análisis Cognitivo Conductual permitiéndonos observar de forma gráfica las relaciones entre las tareas ya encontradas y los distintos objetos en el entorno. Estos grafos también nos permiten conocer el tipo de conocimiento de cada nodo. Lo anterior nos proporciona una riqueza en el sentido de conocer de forma explícita los estímulos provenientes del entorno necesarios para el disparo de las conductas.

La nomenclatura que especifica la relación entre los nodos nos indica lo siguiente:

1. Subclase: implica una abstracción de clases.
2. Antes: implica un acontecimiento que se debe dar antes de otro.
3. Compuesto de: implica la composición de un elemento.

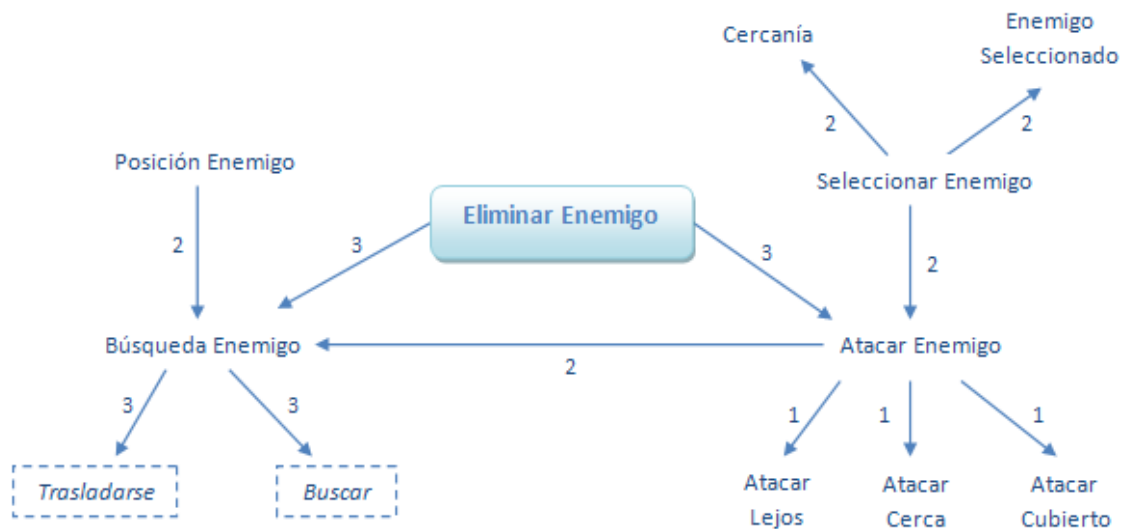


Fig. 5.3 Grafo genético del comportamiento 1

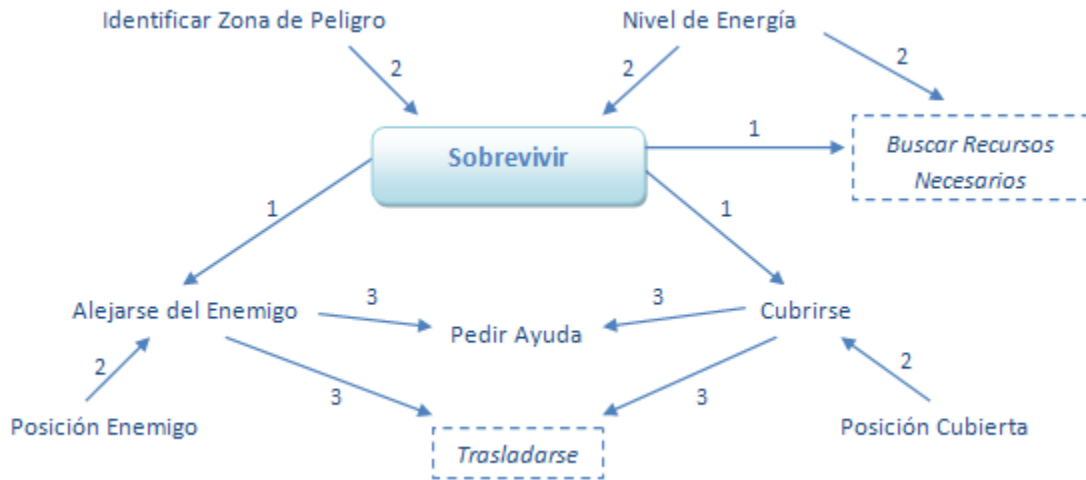


Fig. 5.4 Grafo genético del comportamiento 2



Fig. 5.5 Grafo genético del comportamiento 3

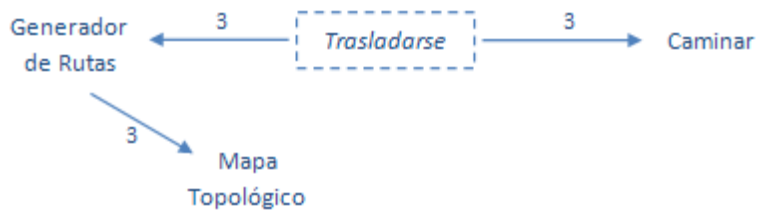


Fig. 5.6 Grafo genético del subcomportamiento trasladarse

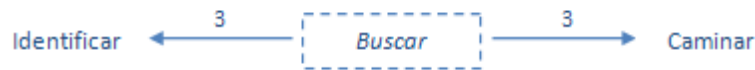


Fig. 5.7 Grafo genético del subcomportamiento buscar

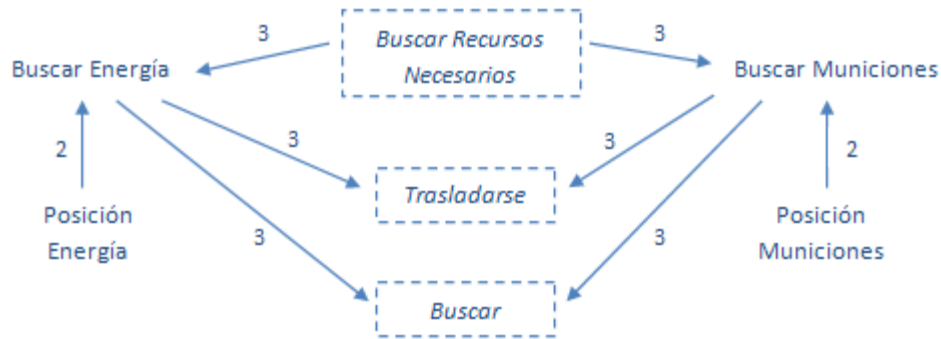


Fig. 5.8 Grafo genético del subcomportamiento buscar recursos necesarios

5.9. El Agente y sus Emociones

Hasta ahora hemos definido el entorno así como esclarecido los comportamientos y las acciones básicas con las cuales los agentes afectarán al entorno para cumplir con sus objetivos. Por otro lado tenemos un modelo emocional que agrupa un conjunto de elementos que permitirán al agente realizar una síntesis de sus emociones. Con todo esto, identificamos un proceso cíclico en el cual, el agente será guiado en parte por sus emociones para cumplir sus objetivos. Para aclarar este proceso describiremos a continuación cada una de sus partes.

El entorno es un mundo virtual dinámico, es decir, conforme pasa el tiempo este actualiza su estado según sus reglas y es alterado por los agentes que interactúan con él. El entorno puede estar representado de muchas maneras, sin embargo, sin importar su representación sabemos que de un tiempo inicial a un tiempo final, un cambio en cualquiera de sus variables indicará una alteración de dicho entorno. La alteración puede interpretarse de diferentes maneras, según sea el caso. Una interpretación sobre un cambio del entorno puede sugerir el movimiento del sol en la bóveda celeste virtual, lo que requerirá de un conjunto de operaciones que permitirán producir una representación gráfica del entorno, es decir, cuando finalmente vemos en la pantalla una imagen dibujada de nuestro mundo virtual siendo iluminado por una fuente de luz.

Una segunda interpretación de los cambios del entorno sugiere el surgimiento de acontecimientos, no presentes hasta el momento, que son de relevancia para los agentes. Dichos acontecimientos contienen información referente a lo que está sucediendo en el entorno, la cual es interpretada por los agentes en busca de patrones específicos. Incluso es posible que estos acontecimientos sean producidos por ellos mismos [LAUA03] y [LAUA05].

Un agente produce cambios en su entorno por medio de sus acciones y comportamientos, los cuales tienen como objetivo el cumplimiento de sus metas mientras son guiados por sus normas. Sin embargo, un agente puede tener un conjunto de metas y normas, cada una de las cuales puede tener uno o más comportamientos asociados los cuales, incluso, pueden ser inhibitorios entre sí, entonces, ¿cómo el agente puede discernir entre el conjunto de metas para elegir el comportamiento más adecuado a la situación? Para responder a esta pregunta imaginemos a un individuo cualquiera, quien probablemente tendrá unas metas a seguir, por ejemplo: comprar un coche, terminar una carrera profesional y tener una familia. Todas estas metas pueden formar una jerarquía según sus preferencias y consciente de su propio estado económico, social y emocional, es decir, que esta persona puede preferir tener una familia antes de comprar un coche pero antes que cualquier cosa desea terminar una carrera profesional. De igual manera un agente en un mundo virtual debe establecer una jerarquía de sus metas [HERE08] con este conocimiento en claro, puede evaluar la situación (información proveniente del entorno) de forma que inicialmente trate de generar un comportamiento que le ayude a completar su meta de mayor jerarquía. Si esta última se cumple entonces, se prosigue con la siguiente meta en la jerarquía y así sucesivamente.

De esta forma podemos establecer para nuestro agente una jerarquía inicial de metas, la cual se muestra en la Figura 5.9, mostrando en la parte superior la meta de mayor importancia.

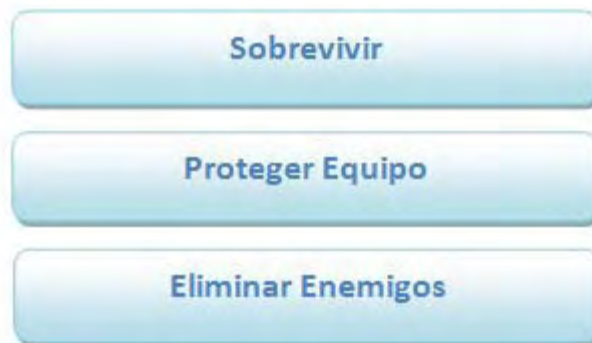


Fig. 5.9 Jerarquía de metas

Es en este punto donde podemos afirmar que nuestro sistema se comportará de la misma manera que un sistema multiagentes tradicional. Es decir, cada agente

persigue sus objetivos seleccionando un comportamiento adecuado con el cual afectara su entorno. Ahora el siguiente paso es incorporar nuestro modelo de emociones.

Si pensamos por un momento en un ser humano, obviamente sabemos que éste puede experimentar un amplio rango de emociones como felicidad, tristeza, enojo, miedo, etc., sin embargo, puede existir un momento en el que simplemente no esta experimentando ninguna de estas emociones, es decir, no esta feliz, triste, molesto, temeroso, etc. Digamos entonces que este ser humano se encuentra en un estado emocional *neutro*. La relevancia de un estado emocional neutro subyace al hecho de que en el caso de que una persona, contará con el tiempo necesario, sería capaz de analizar cada decisión desde distintas perspectivas que le permitirán alcanzar cada una de sus metas. Lo anterior implicaría que la selección estaría muy cerca de la solución óptima para cada una de ellas, además sin ser influenciado por algún parámetro externo o interno. Sin embargo, en una situación más realista sabemos que una persona, rara vez se encontrara conforme con la solución optima y es común pensar en expresiones tales como “y si hubiera...” o “me sentiría mejor si...”.

En [LOEG03] se describe un modelo que explica la influencia de las emociones en el proceso de toma de decisiones, considerando los aspectos positivos y negativos. Simon postula en [SIMH67] que las emociones fijan la atención a eventos importantes. Viendo el sistema nervioso central como un procesador de información serial, Simon supuso que las emociones sirven como *interruptores cognitivos* que nos dicen qué es importante, con el fin de facilitarnos la priorización de los recursos.

Llevando estas ideas a nuestro agente, inicialmente podemos suponer que éste se encuentra en un estado emocional neutro, lo que le permite establecer una jerarquía específica de sus metas, justo como se clarificó anteriormente. Ahora bien, aplicando nuestro modelo emocional, cada acontecimiento percibido por nuestro agente será llevado a un proceso de *valoración*, donde se interpretarán los efectos que este suceso pueda provocar en los planes para alcanzar una meta determinada. Esta valoración producirá un grado de intensidad para cada emoción basado en la importancia de cada meta, es decir, que tan alta se encuentra en la jerarquía. Este parámetro es visto por los autores de la teoría OCC como la variable de *deseabilidad*; mientras más deseable sea (o indeseable), podrá generar una emoción más intensa. Sin embargo, dado que esta valoración puede afectar a un grupo de emociones, ¿cómo sabemos qué emoción esta experimentando el agente? Anteriormente, en este trabajo se formalizo una descripción de las emociones permitiéndonos listar sus propiedades, ahí se identifico un parámetro llamado *umbral*, el cual nos permite de manera precisa afirmar que cuando la intensidad de una emoción supera su umbral entonces la emoción se encuentra *activa*, y por lo tanto, el agente la esta experimentando. Es válido plantear la posibilidad de tener dos o más emociones activas, lo que incluso podría generar un conflicto que usualmente llamamos

emociones encontradas, sin embargo, con el fin de acotar este trabajo simplificaremos esta estructura estableciendo la convención de que entre dos o más emociones activas, se considera la emoción activa aquella que presente la intensidad relativa más alta, esto es, la que superó por mayor cantidad su umbral.

Ya que podemos afirmar que nuestro agente está sintetizando una emoción determinada, ahora mostraremos cómo es afectado su comportamiento. Inicialmente identificamos tres áreas donde las emociones pueden influir. La *percepción* y la *ejecución* se encuentran en el extremo inicial y final del sistema que representa al agente. En la etapa de entrada (inicial) de los datos, de acuerdo a la percepción del agente, éste puede ver con más claridad o sentido la información actual. En el caso de la información que sale (etapa final) del sistema que se traduce en las acciones que realiza el agente para afectar su entorno, éstas pueden verse alteradas por un filtro de ejecución, lo que se traduce en una alteración de la forma de las acciones pero sin cambiar su función final. La etapa intermedia, causa de la atención de nuestro trabajo, es el proceso de *toma de decisiones*, la selección del plan más adecuado para la meta que persigue el agente. En este caso particular las emociones tendrán como consecuencia la alteración de la jerarquía de metas. En otras palabras, después de la valoración de los acontecimientos y dada la aparición de una emoción activa, la estructura jerárquica de metas sufre una reorganización basada en: a) el estado actual del agente, b) su percepción del mundo, y c) su estado propioceptivo. Para el agente, puede significar que después de la presencia de la emoción de estar contento su estructura de metas podría reorganizarse como se muestra en la Figura 5.10.



Fig. 5.10 Reorganización de la jerarquía de metas.

Finalmente, dado que el agente ahora cuenta con una jerarquía de metas diferente deberá ajustar su estado interno de manera que refleje la necesidad de cumplir con su meta primaria y subsecuentemente con las otras de menor nivel. Esta última etapa se observa en el agente al presentar un comportamiento acorde con el cual afectará el entorno para iniciar el ciclo nuevamente.

En conclusión, nuestra propuesta plantea un ciclo de interacción mostrado en la Figura 5.11 y que se resume de la siguiente manera:

Los acontecimientos producidos en el entorno son interpretados por los agentes que realizan una valoración con base en la deseabilidad o jerarquía, la cual produce incrementos en la intensidad de la emoción. Si esta última se presenta genera una reorganización en la jerarquía de las metas, y por lo tanto, una reevaluación del comportamiento actual. El comportamiento se traduce en acciones que afectan el entorno produciendo cambios que eventualmente producirán un acontecimiento.



Fig. 5.11 Ciclo de la Arquitectura Emotiva



Capítulo 6

Desarrollo de la Plataforma de Experimentación

Hasta ahora a lo largo de este trabajo, se han introducido distintos aspectos teóricos y el diseño inicial de un sistema que permitirá a personajes virtuales actuar en base a un modelo emocional. Como parte importante de la confirmación de nuestra teoría se desarrollo un sistema que permitirá experimentar con la arquitectura afectiva y su influencia en los agentes. Este sistema se implementó con las características y restricciones de un videojuego, lo que en muchos sentidos es importante para la selección de las herramientas además de las técnicas que se utilizarán.

Aunque existen videojuegos que exponen el código de su implementación libremente, la decisión de crear nuestra propia implementación fue guiada por la necesidad de tener un completo control sobre cada aspecto del juego, tanto en la manipulación de los eventos del entorno como en la conducta de los personajes.

6.1. Herramientas de Desarrollo

6.1.1. XNA

XNA es un conjunto de herramientas creadas por Microsoft Corp. con el propósito de facilitar el desarrollo y el mantenimiento de videojuegos, aunque tiene como audiencia principalmente a estudiantes y aficionados, esta tecnología ha demostrado exitosamente su capacidad para la creación de prototipos o de videojuegos sencillos tipo *arcade*. Incluso al día de hoy se han realizado

competencias apoyadas por Microsoft donde los videojuegos ganadores se someten a un proceso de evaluación para finalmente convertirse en productos comercializados a través de la red *Xbox Live Arcade*.

Aunque pueden encontrarse en la red una gran cantidad de herramientas para el desarrollo de videojuegos, XNA ha alcanzado rápidamente una gran popularidad y maduración, pues al ser apoyada por una de las empresas más grandes de software cuenta con una amplia comunidad, y tecnologías a su disposición. El uso de el lenguaje de programación C#, incluyendo el ambiente de desarrollo integrado Microsoft Visual Studio tanto en su versión profesional, como en su versión gratuita, ofrecen a los desarrolladores un entorno conocido, probado y robusto para sus proyectos. Por otro lado, XNA es la única herramienta de bajo costo disponible a la comunidad que permite la programación de una consola comercial de videojuegos de última generación, es decir, la propia consola de Microsoft, el Xbox 360. Esta opción ofrece un hardware de alto desempeño con un procesador de tres núcleos con dos hilos por hardware en cada uno, lo que se traduce en algoritmos más complejos corriendo simultáneamente lo que a su vez ofrece la posibilidad para juegos de alta calidad aún sin la programación de bajo nivel que se ofrece en los kits de desarrollo profesional.

6.1.2. Arquitectura de XNA

Como se mencionó anteriormente, XNA es un conjunto de herramientas que permiten el desarrollo de videojuegos en distintas plataformas. El componente principal de desarrollo de esta arquitectura es llamado *XNA Game Studio (XGS)*, que al momento de la realización de este trabajo se encuentra en su versión 3.0. En la Figura 6.1 pueden observarse los distintos componentes que lo conforman, donde en el nivel más alto encontramos a *Microsoft Visual Studio* ofreciendo un ambiente de desarrollo por medio de C#, también en su versión 3.0, por debajo de este tenemos el *XNA Framework* que proporciona el conjunto de bibliotecas exclusivamente para facilitar el desarrollo de un videojuego. En el siguiente nivel encontramos el *.net Framework* y *.net Compact Framework* que proporcionan el conjunto de interfaces básicas que cada plataforma de hardware puede ofrecer, las cuales pueden observarse en el nivel inferior, ofreciendo al momento cualquier computadora personal con el sistema operativo Microsoft Windows, la consola Xbox 360 y el reproductor digital Zune.



Fig. 6.1 Componentes de XNA Game Studio

6.1.3. XNA Framework

Para entender porque XNA facilita el trabajo del desarrollador es necesario describir las partes que conforman el *XNA Framework* y que ofrece cada una de ellas. El *XNA Framework* esta formado por cuatro capas, como se observa en la Figura 6.2, cada una de ellas agregando funcionalidad a las anteriores ofreciendo componentes reusables, extensibles y un modelo inicial de aplicación:

- *Plataforma*. En la base de la arquitectura se encuentran diferentes bibliotecas e interfaces que se enlazan con el hardware objetivo que se este desarrollando. Esta capa implementa las capacidades especificas del hardware, pues evidentemente el potencial de procesamiento es distinto entre una consola de videojuegos dedicada a un dispositivo portátil de bajo consumo de energia. Por lo tanto, ya que cada hardware tiene distintas capacidades y limitaciones, esta capa se encarga de exponer la funcionalidad especifica necesaria al proyecto.

- *Core Framework.* Esta capa ofrece un conjunto de bibliotecas que proveen una arquitectura homogénea y común al desarrollo de videojuegos, de esta forma se exponen interfaces para: el dibujo en pantalla, reproducción de audio, procesamiento de datos de entrada, almacenamiento, servicios de red y funciones matemáticas de uso frecuente. De esta manera el mismo código usado para dibujar una imagen en pantalla será idéntico no importando la plataforma destino, debido a que la tarea de implementación específica se traslada a la capa inferior, es decir, la capa de Plataforma.
- *Extended Framework.* Esta capa proporciona al desarrollador un modelo de aplicación que implementa la funcionalidad de un juego, es decir, ofrece el ciclo básico presente en lo todo videojuego que consiste en: inicialización de componentes, procesamiento de datos de entrada, dibujo de la representación de juego y procesamiento de la lógica del juego. Además el modelo de aplicación ofrece la implementación de servicios y componentes, con el fin de generar bloques independientes que agreguen funcionalidad específica pudiendo ser reutilizados en distintos proyectos. Adicionalmente, esta capa ofrece el Pipeline de contenido que permite procesar los distintos formatos de archivos digitales, como imágenes, modelos tridimensionales, sonido, etc. y procesarlos en un formato optimizado para la plataforma, con la ventaja de que si un formato específico no es soportado, el desarrollador tiene la opción de extender este componente para incluir su contenido de forma transparente a la aplicación final.
- *Juegos.* Finalmente, la última capa consiste en la aplicación final generada por el desarrollador del juego mismo. Componentes diversos creados por la comunidad pueden adoptarse para agregar funcionalidad al código creado por el programador que a su vez utilice el contenido multimedia creado por artistas. De esta forma, muchas de las tareas no específicas de la implementación quedan relegadas a capas inferiores facilitando en gran medida la producción final de un videojuego.

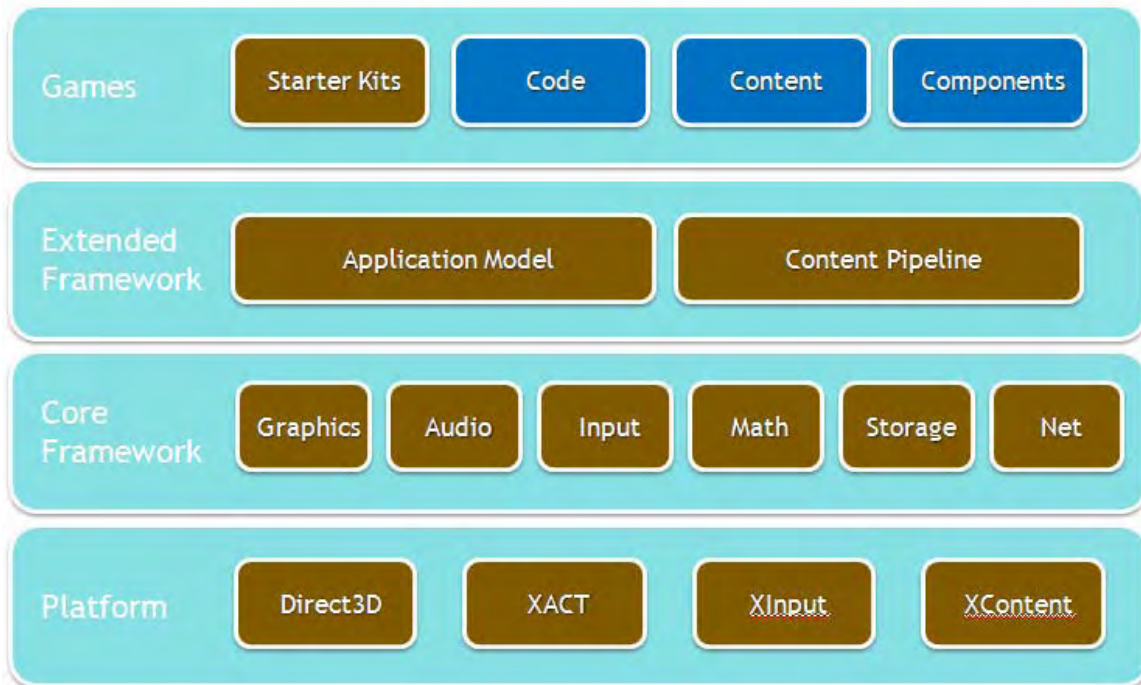


Fig. 6.2 XNA Framework

Con la decisión de adoptar XNA como nuestra herramienta, en las siguientes secciones de este capítulo nos enfocaremos a describir la implementación específica de nuestro videojuego, identificando parte a parte los distintos componentes que lo conforman, así como su funcionamiento e interacción y como XNA o las distintas bibliotecas basadas en XNA ofrecen la funcionalidad requerida para nuestro sistema.

6.2. Arquitectura de la Implementación

En este apartado se describe el esquema básico sobre el cual trabaja nuestro sistema, proporcionando las distintas pantallas necesarias para un videojuego así la administración y comunicación entre los distintos objetos que representan el mundo virtual generado.

6.2.1. Administración del flujo del juego

Dado que nuestro sistema está implementado como un videojuego, es importante que este cuente con los elementos comunes a estos, no solo para ofrecer una interfaz homogénea sino para simplificar su uso y permitir a los jugadores una interacción sencilla y amigable. Para esto se requiere de un sistema que administre la transición entre pantallas, así como la actualización y

el dibujo de cada pantalla individual. De esta forma tomando como base una implementación propuesta en XNA Creators Club Online [XCCO09] llamada *Game State Management*, implementamos un sistema de pantallas que nos facilita la transición entre cada una de ellas, generando su lógica y contenido de manera independiente. Las clases que componen el flujo del juego se muestran en la Figura 6.3

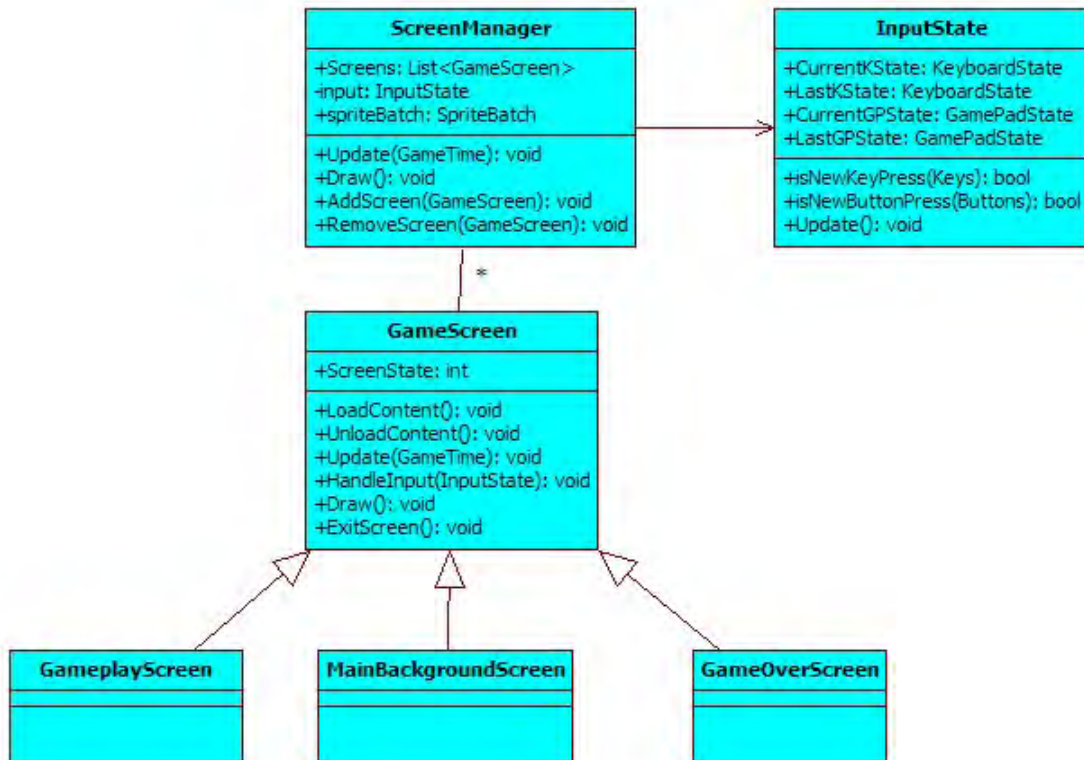


Fig. 6.3 Clases para la administración del flujo del juego

La clase *ScreenManager* está implementada como un componente dentro del paradigma de XNA. Su función es administrar la presentación, superposición y actualización de las pantallas, así como determinar cuál de ellas es la pantalla activa. Además incluye una instancia de la clase *InputState* la cual permite hacer una abstracción de toda señal de entrada, ya sea teclado o gamepad, la cual podrá ser usada por cada pantalla que herede de la clase *GameScreen*, con la que se proporcionará la funcionalidad básica de inicialización, actualización, manejo de datos de entrada y dibujo.

6.2.2. Jerarquía de Entidades

En nuestro mundo virtual, una *entidad* puede definirse como un objeto con ciertas propiedades y funcionalidad. Cada entidad es diferenciada por un identificador único a lo largo de todo el sistema con el cual puede obtenerse una referencia con el propósito de transformarla de alguna manera. Debido a que cada objeto en el juego posee una lógica distinta, cada uno de ellos debe heredar sus capacidades de clases especializadas, con el propósito de administrar de manera eficiente el sistema permitiendo la manipulación de cientos de objetos simultáneamente. En la Figura 6.4 se muestra la organización de dichas clases.

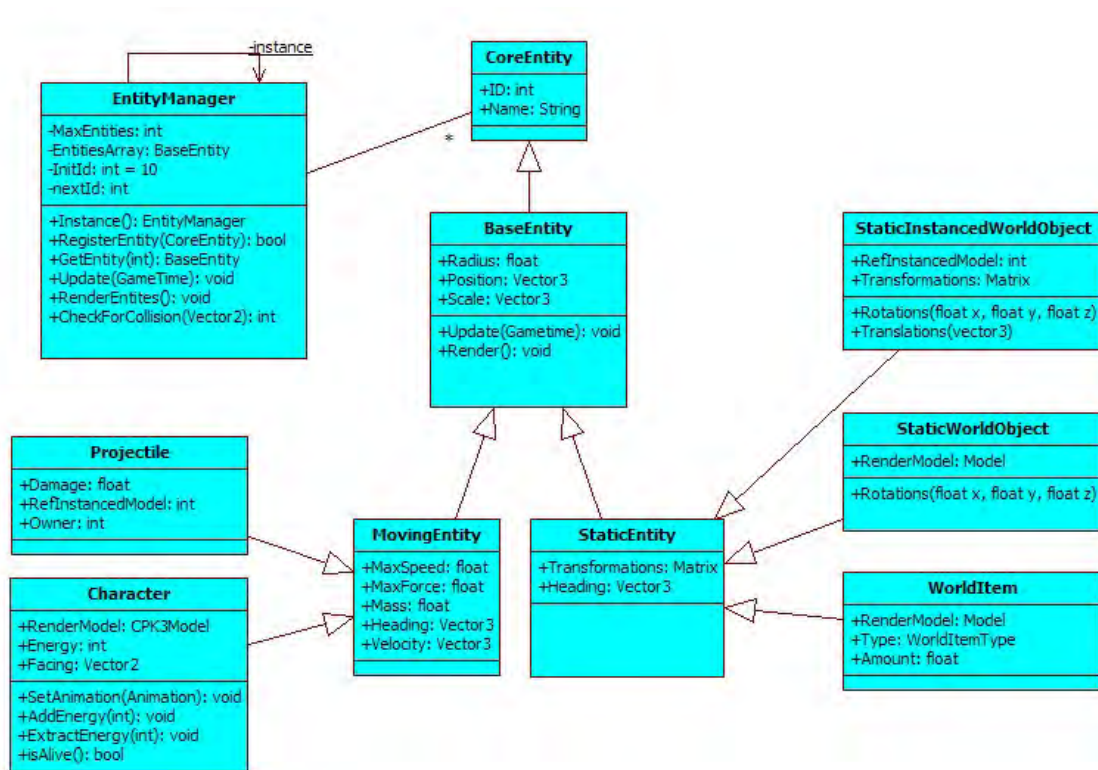


Fig. 6.4 Entidades en el sistema

Las clases *CoreEntity* y *BaseEntity* proporcionan la funcionalidad inicial a cada uno de los objetos ofreciendo entre otras cosas: un identificador único, nombre, posición en el mundo, escala, radio de cobertura, así como los métodos *Update* y *Render* los cuales cada subclase debe sobrescribir proporcionando la implementación específica de la lógica interna y el dibujo del objeto en pantalla.

Para objetos estáticos del entorno, es decir, objetos que no cambiaran su posición a lo largo de la ejecución del sistema, se proporciona la clase *StaticEntity* de la cual heredan las clases *WorldItem*, *StaticInstancedWorldObject*

y *StaticWorldObject* que cumplen con la tarea de representar respectivamente ítems que pueden recogerse, conjuntos de objetos con un modelo tridimensional único y objetos que forman el escenario del mundo virtual.

Por otro lado, los objetos basados en la clase *MovingEntity* están diseñados para cambiar su posición continuamente a lo largo de la ejecución del sistema. Es en estas clases donde el método *Update* resalta su importancia al realizar las operaciones necesarias para calcular la nueva posición del objeto. Debido a la naturaleza de nuestro videojuego, existen dos tipos de objetos que requieren de esta funcionalidad, el primero de ellos son los personajes mismos uno de los cuales es el avatar del usuario, implementado en la clase *Character*, y en segundo lugar, los proyectiles disparados por los propios personajes son abstraídos usando la clase *Projectile*, la cual les ofrece propiedades como el daño que pueden causar, o la entidad que fue responsable de su actividad, es decir, quien disparó este proyectil.

Finalmente, se requiere de un monitor que administre la creación y mantenimiento de cada una de las entidades presentes en el videojuego. La clase *EntityManager*. Una de las más importantes del sistema se encarga de registrar cada entidad para su uso, actualizando y dibujando a cada una de ellas en cada ciclo del sistema, así como proporcionando métodos que permiten obtener referencias a las propias entidades de una manera eficiente.

6.2.3. Modelo de Mensajes

Mat Buckland en [BUCM05] nos afirma que videojuegos bien diseñados tienden a ser manejados por eventos, es decir, cuando sucede algún evento en nuestro mundo virtual, ya sea que un arma es disparada o un enemigo es derrotado, el evento es enviado a todos los objetos en el juego de forma que puedan responder apropiadamente. De esta forma, contamos con un sistema de comunicación entre todas las entidades del videojuego, donde su eficiencia radica en que los objetos deben esperar la recepción del mensaje a diferencia del sensado continuo del entorno en busca de cambios específicos.

La clase *Message* empaqueta la información relevante de los eventos sucedidos en el entorno para ser enviados a las entidades destino, las cuales deben implementar la interface *IMessageReceiver* que define únicamente el método *HandleMessage* con el cual los objetos recibirán el mensaje y responderán según su lógica interna.

Un objeto instanciado de la clase *Message* contiene identificadores de la entidad que envía el mensaje y el destinatario, también incluye su tipo y el tiempo de entrega, parámetro de mucha utilidad cuando se requiere mandar un mensaje no de manera instantánea sino con cierto retraso de tiempo. Por último, una variable adicional permite incluir información extra referente al evento específico.

Para que este sistema pueda funcionar debe existir un mensajero que entregue cada uno de los mensajes generados. La clase *MessageDispatcher* es la encargada de esta tarea, proporcionando una interfaz donde cada objeto estructura su mensaje y lo envía usando el método *DispatchMessage*, verificando su tiempo de envío para, posteriormente, obtener una instancia de la entidad destino por medio de *EntityManager* y finalmente entregar el mensaje.

La Figura 6.5 muestra un diagrama de las relaciones existentes entre estas clases.

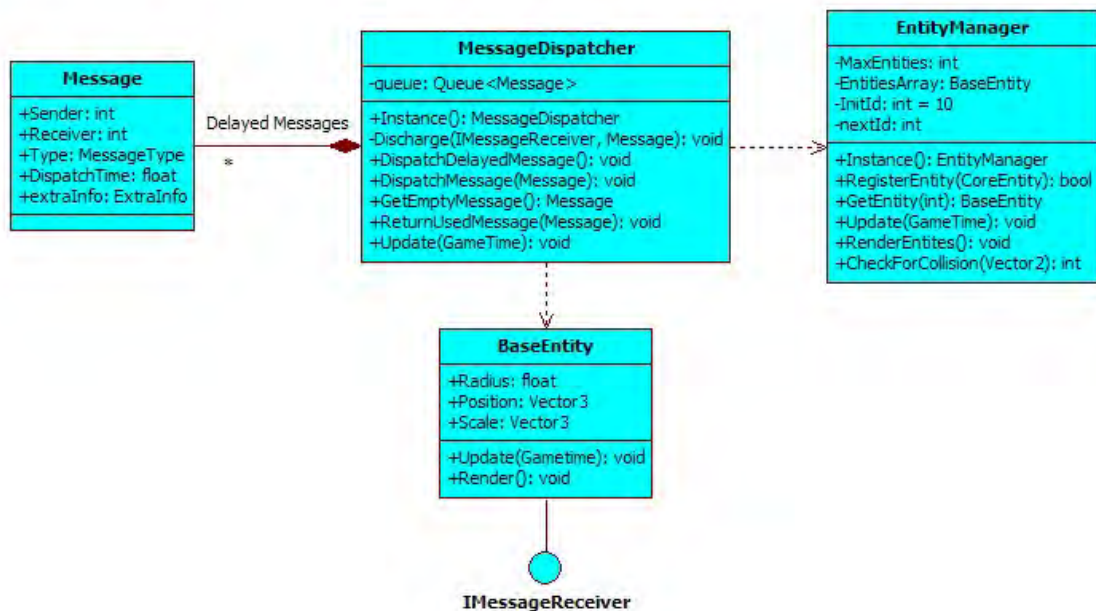


Fig. 6.5 Modelo de Mensajes

Cada mensaje se identifica primariamente por su tipo, los cuales pueden agruparse en distintas categorías dependiendo de su objetivo, algunos son creados para interpretar los datos de entrada por parte del usuario, para activar comportamientos específicos o simplemente para informar sobre algún suceso como es la colisión con otro objeto.

Algunos de los mensajes que podemos identificar en nuestro sistema son:

- *Msg_User_Move*. Indica que el usuario solicito el movimiento de su avatar proporcionando la cantidad de desplazamiento en el plano.
- *Msg_User_Shoot*. Indica que el usuario solicitó que su avatar dispare su arma actual en la dirección actual.

- *Msg_Behavior_Seek_On*. Indica a la entidad que debe activar su comportamiento que lo guiará hasta un punto dato.
- *Msg_ReceiveDamage*. Significa que la entidad que recibe el mensaje fué golpeada por otra y ésta debe responder, generalmente, disminuyendo su energía de manera adecuada. Si la energía llega a cero indicaría que la entidad esta inactiva y no puede moverse lo que puede generar un nuevo grupo de mensajes.
- *Msg_CreateProjectile*. Cada vez que un personaje debe disparar su arma envía este mensaje con su posición y dirección a la clase correspondiente, para generar de manera eficiente un objeto que representa a cada proyectil, el cual estará activo a partir de ese momento hasta colisionar con otro objeto en el entorno.
- *Msg_AwakeEnemy*. Cada vez que un enemigo es eliminado se genera este mensaje que es entregado con un determinado retraso de tiempo, pues provocará que el enemigo se encuentre activo de nuevo para seguir interactuando en el entorno.
- *Msg_Item_Taken*. Generado por cualquier personaje, ya sea autónomo o controlado por el usuario, indica a un ítem que el personaje intenta obtener beneficios de él.
- *Msg_Item_ReceiveAmmo*. Cuando un ítem del tipo que representa las municiones del arma recibe el mensaje de que otra entidad trata de obtener beneficios de él, éste envía un mensaje indicando que puede incrementar sus municiones en cierta cantidad determinada por el ítem mismo.
- *Msg_Soc_HelpSignal*. Al tratar con un grupo de agentes suponemos cierta interacción entre ellos, así este mensaje representa un tipo de comunicación social que les indica que un personaje necesita ayuda y ha enviado una señal a sus compañeros para que conozcan su situación.
- *Msg_Emotion_FriendlyAttack*. Claramente, podemos crear mensajes relacionados con las emociones y sus eventos relacionados. De forma que este mensaje es enviado cuando se detecta que un personaje ha atacado a uno de sus compañeros de alguna forma. Lo cual evidentemente, puede ocasionar una ola de mensajes que puede concluir en la interpretación de acontecimientos, reorganización de metas y activación de emociones.

6.2.4. Cola de Objetos

Aún con las ventajas que el uso de la plataforma XNA nos provee, encontramos ciertos paradigmas relacionados con la eficiencia del sistema debido al uso de un lenguaje interpretado como es C#.

La causa principal del decremento de eficiencia es el *Garbage Collector* (GC) [MCGC09], el cual maneja la asignación y liberación de memoria en el sistema. Así con la creación de cada objeto se asigna un bloque de memoria que cuando ya no es utilizada por la aplicación este se convierte en *basura* (o memoria inutilizable) mientras es liberada por el GC. Dado que este proceso interrumpe el flujo natural de sistema y el programador no tiene control sobre esto, la única forma de mejorar la eficiencia es minimizar la creación de nuevos objetos.

En nuestra implementación identificamos dos fuentes causantes de este problema: la generación de mensajes y la creación de proyectiles disparados por los personajes. Bettcher [BETT08] propone una solución que permite reutilizar objetos colocándolos en una estructura de la cual pueden extraerse para su uso. Adaptando esta idea, utilizamos una estructura tipo *FIFO*, la cual inicialmente es inicializada con instancias de nuestros objetos, tanto del tipo *Message* como *Projectile*. Así cuando una entidad requiera de alguno de estos objetos sólo será necesario extraer el siguiente elemento de nuestra cola, y al terminar su uso bastará simplemente con agregarlo al final de la cola para que otra entidad pueda hacer uso de él nuevamente.

6.2.5. Modelo de Colisiones

Uno de los subsistemas básicos que nuestro videojuego debe incorporar es un sistema de colisiones, con el cual podrá detectar el momento en que dos entidades pretendan ocupar el mismo espacio en el entorno. En un estilo de juego como el propuesto esto es de vital importancia, pues es necesario tener un mecanismo que nos indique la colisión entre dos entidades, de lo contrario, podríamos observar personajes que traspasan las paredes sin ninguna dificultad. Por otro lado, sin un sistema de colisiones no tendríamos manera de saber cuando un personaje es alcanzado por un proyectil, lo que rompería por completo con el esquema del juego pues ninguno de ellos morirá.

Desafortunadamente, la gran cantidad de objetos presentes en los videojuegos actuales nos obligan a contar un modelo eficiente. Por ejemplo en nuestro caso, el cual sabemos no es uno de los más complejos, cada vez que un proyectil es disparado, este debe verificar la colisión contra algún objeto o personaje, en caso de una evaluación positiva terminar su movimiento y enviar los mensajes apropiados. Poniendo algunos números si suponemos que tenemos en pantalla alrededor de 15 proyectiles disparados por 5 personajes en un entorno con 195 objetos estáticos con los cuales puede colisionar, entonces en cada ciclo del

sistema tendríamos que realizar alrededor de 3000 pruebas únicamente para los proyectiles, algo que evidentemente es muy costoso aún con el algoritmo más simple.

Para solucionar este problema implementamos dos esquemas que nos permiten reducir la carga de trabajo los cuales son descritos a continuación.

6.2.5.1. Octree

La primera optimización que podemos aplicar a este problema es la partición del espacio del entorno utilizando una estructura llamada *Octree* [GINS00]. Mostrada en la Figura 6.6 un *octree* se define como una estructura tipo árbol en donde cada nodo interno tiene ocho nodos hijos, en el caso de un espacio tridimensional cada nodo representa una partición en ocho octantes iguales.

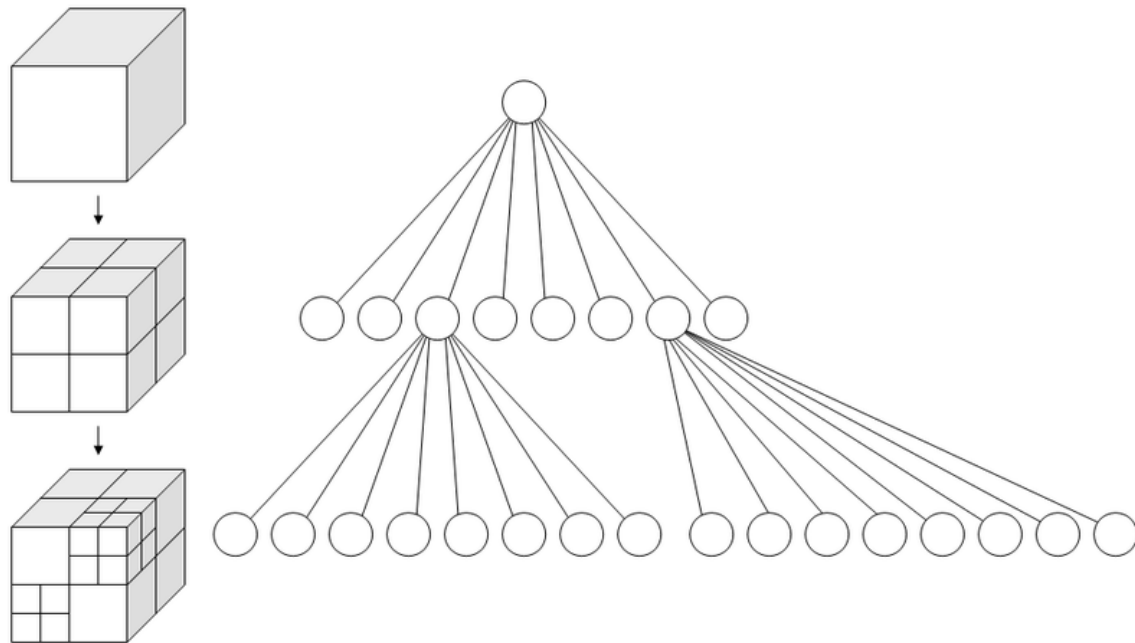


Fig. 6.6 Representación de un octree

Con la ayuda de un *octree* cada entidad tiene su posición en el interior de alguna de las particiones más pequeñas, al actualizar su movimiento verifica los octantes vecinos y ubica su nueva posición. Así el sistema de colisión puede utilizar un *octree* para obtener una referencia de las entidades que se encuentren en el mismo octante, las que serán significativamente menos que la totalidad de ellas.

En nuestro sistema este esquema está implementado por medio de las clases *Octree* y *OctreeNode*, las que proporcionan métodos para introducir referencias

de las entidades así como obtener una lista de éstas en cada uno de los nodos finales.

6.2.5.2. Colisión mediante esferas

Ya que contamos con una partición espacial de nuestro entorno y esto nos proporciona una lista reducida de entidades, debemos entonces realizar algún tipo de prueba con la finalidad de detectar la presencia de una colisión entre entidades.

Para esta prueba, utilizaremos el concepto de *esfera delimitadora* o *bounding sphere*, que se define como la esfera de mínimo tamaño que puede encerrar en su totalidad un objeto, en este caso, un modelo tridimensional. Cada una de las entidades que heredan de la clase *BaseEntity*, contienen la información necesaria para definir una estructura de este tipo.

El algoritmo para detectar una colisión usando esferas delimitadoras, consiste en calcular la distancia existente entre los centros de dos esperas y si ésta es menor a la suma de ambos radios entonces existe una colisión entre ellas. La ventaja de esta aproximación es que requiere de muy pocos cálculos para obtener un resultado, aunque su desventaja es que su precisión no es muy buena y con frecuencia se obtienen falsos positivos. XNA ofrece clases y métodos que utilizan este modelo lo cual facilita su implementación en nuestro proyecto.

Para los propósitos de este trabajo, este modelo es suficiente para alcanzar un nivel aceptable entre rendimiento y exactitud de las colisiones, en caso de requerir un nivel más de detalle, este puede implementarse a partir de la detección de una colisión entre esferas delimitadoras lo cual permite mantener el rendimiento del sistema.

6.3. Dibujo de la Escena

Cada videojuego consta de una etapa donde la representación de su estado interno es mostrada al usuario por medio de un dispositivo como un monitor de computadora o televisor. El videojuego propuesto incorpora una representación de un mundo virtual tridimensional, en el cual se analizan todos los objetos presentes y después de una serie de transformaciones estos pueden expresarse en píxeles individuales que serán mostrados en cada ciclo de dibujo del videojuego.

La comunidad de XNA e incluso la propia implementación de Microsoft, proporciona una gran cantidad de opciones para satisfacer esta tarea. A continuación, se describen brevemente algunas de las utilizadas por este

proyecto con la finalidad de presentar un videojuego con una buena calidad gráfica.

6.3.1. Modelo de Cámara

Un modelo de cámara es un conjunto de parámetros matemáticos que describen la forma en como un objeto tridimensional es transformado para obtener una representación bidimensional que pueda ser mostrada en una pantalla. Dado que estos parámetros presentan una abstracción de una cámara en el mundo real está puede modificarse a partir de variables como: la posición de la cámara, su orientación, etc.

A continuación se describen los dos modelos de cámara utilizados por este proyecto.

6.3.1.1. Cámara Mirar hacia

El principal parámetro de esta cámara identifica la dirección hacia la que esta orientada, de forma que al cambiar la posición de ésta su orientación es ajustada para seguir enfocando el mismo punto. Esto nos ayuda a mantener en pantalla al personaje controlado por el usuario en todo momento no importando el movimiento de este.

Esta cámara esta implementada por medio de la clase *LookAtCamera*, que ofrece la propiedad para ajustar el vector que indica la posición hacia donde mira. Así en cada actualización del movimiento del personaje se ajusta la orientación de la cámara como sigue:

```
CurrentCamera.LookAtCamera = Avatar.Position;
```

6.3.1.2. Cámara libre

Esta cámara ofrece una libertad de movimiento absoluta donde se puede asignar no solo el vector de posición sino también la rotación en cada uno de los ejes. Su implementación esta basada en *cuaterniones* [EBED06], lo que evita problemas como el *gimbal lock* que produce rotaciones no deseadas.

La cámara libre se implementa con el objetivo de mostrar una vista general de toda la escena y permitirnos evaluar las acciones que suceden en el entorno. Implementada por la clase *QuaternionCamera*, ésta proporciona métodos para trasladarse y rotar en cualquiera de los ejes.

6.3.2. High Level Shading Language

High Level Shading Language o simplemente *HLSL* es un lenguaje de alto nivel, que permite la programación del pipeline gráfico en plataformas basadas en *DirectX* desde su versión 8. Cada programa escrito en HLSL es llamado *shader* y define la representación visual final de los objetos en pantalla. Los datos que forman cada objeto a dibujarse pasan como un flujo de primitivas que son procesadas por dos etapas:

- *Vertex Shader*. Realiza operaciones a nivel de vértices como puede ser transformaciones, skinning, desplazamiento de vértices, etc. Como mínimo, esta etapa debe dar por resultado la posición de cada vértice, sin embargo, puede generarse información adicional de utilidad para etapas posteriores.
- *Pixel Shader*. Recibiendo información proveniente del *vertex shader*, esta etapa realiza operaciones a nivel de píxeles como combinación de texturas, cálculo del modelo de iluminación, etc.

XNA nos permite manipular enteramente el aspecto visual de nuestra aplicación por medio de la clase *Effect*, la cual encapsula la funcionalidad de un *shader* escrito en HLSL.

A continuación, se describen algunos de los *shaders* utilizados por nuestra aplicación

6.3.2.1. BasicEffect

BasicEffect es una clase derivada de *Effect*, la cual proporciona un *shader* básico con soporte para color en vértices, texturas e iluminación. Este es uno de los ejemplos más claros de lo que el XNA Framework ofrece a los programadores para facilitar el desarrollo del juego, pues cada modelo tridimensional cargado en memoria cuenta con su propia instancia de *BasicEffect*, y bastarán unas cuantas líneas de código para mostrar su modelo en pantalla sin tener que escribir ningún *shader* en HLSL.

La Figura 6.7 muestra la calidad visual que presenta un modelo dibujado en XNA por medio del *BasicEffect*.



Fig. 6.7 Dibujo de un modelo usando *BasicEffect*

Debido a su simplicidad y calidad visual este proyecto utiliza este efecto para dibujar la mayoría de los objetos en pantalla, dejándonos la tarea de generar los modelos y texturas necesarios representativos de cada objeto en la escena.

6.3.2.2. InstancedEffect

Nuestro entorno virtual esta formado por un conjunto de objetos que representan un ambiente sobre el cual los personajes interactúan. Objetos que simulan cajas forman un área de juego con obstáculos que diversifica las situaciones que pueden presentarse, sin embargo, dibujar tantos objetos en pantalla puede ser un proceso costoso e incluso prohibitivo que es aún peor si tomamos en cuenta que nuestro sistema cuenta con múltiples instancias de proyectiles causando una sobrecarga similar al proceso de dibujo. Esto es un problema común que existe en los videojuegos actuales, por lo que existen distintas formas de resolverlo. En nuestro caso, adoptaremos una técnica llamada *Instanciación* implementada en el ejemplo *Mesh Instancing* [XCCO09].

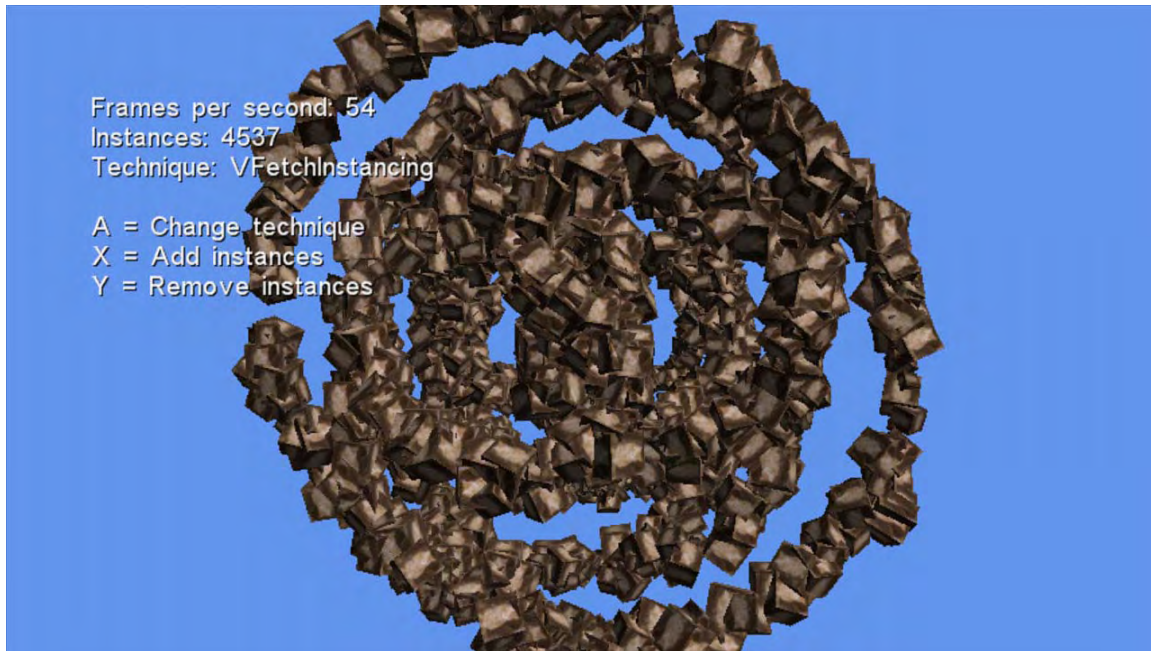


Fig. 6.8 Dibujo de miles de objetos por medio de la técnica Instanciación

Como se observa en la Figura 6.8 es posible dibujar con esta técnica miles de objetos iguales sin perder el rendimiento y manteniendo un framerate constante. Aunque la implementación puede variar dependiendo el hardware disponible, básicamente la instanciación consiste en enviar al pipeline gráfico no solo los datos que forman la malla sino también un arreglo de transformaciones, con las que podrá dibujar el mismo objeto repetidamente aplicando los cambios en el GPU y disminuyendo la carga del CPU.

Una de las clases que implementan esta aproximación en nuestro sistema es la llamada *StaticInstancedWorldObject*. Dentro del juego se genera un arreglo de estos objetos cada uno de los cuales contiene una de las transformaciones que se aplicarán a un modelo externo referenciado por un identificador dentro del mismo objeto.

6.3.3. Preprocesamiento y posprocesamiento

La calidad visual de los videojuegos actuales es debida no a una técnica en particular sino a un grupo de elementos que aunque simples individualmente, en conjunto muestran una imagen más rica llena de detalles que serán de importancia para que el usuario perciba un ambiente más real, aún en situaciones ficticias y con un acabado más profesional, lo cual permitirá que el aspecto visual no sea un factor de distracción importante que evite la evaluación de nuestro modelo emocional.

6.3.3.1. Skybox

Cuando representamos una escena, generalmente, esta se encuentra ubicada en un entorno cerrado como una habitación, o en un entorno abierto por ejemplo, un prado a la luz del medio día. El entorno de nuestro videojuego se encuentra situado en una locación a cielo abierto lo que carga con el problema de modelar todo un ambiente que involucre objetos hasta donde la vista puede alcanzar. Este puede ser un trabajo arduo para el artista, pero además agrega una sobrecarga de información que puede evitarse con el uso de un *skybox*.

Un *skybox* genera una ilusión de un ambiente abierto colocando imágenes bien diseñadas en las caras internas de un cubo proporcionando una imagen constante en todas direcciones. En la Figura 6.9 se aprecia que la ilusión es producida al introducir al cubo los objetos así como a la cámara con lo que se consigue un aspecto de profundidad y espacio abierto. Su eficiencia es bastante alta debido a la baja carga que produce el cubo con únicamente ocho vértices que procesar. El uso de un *skybox* se considera una etapa de preprocesamiento de la imagen, debido a que esta debe usarse como fondo sobre el cual se dibujará todos los demás objetos de la escena.

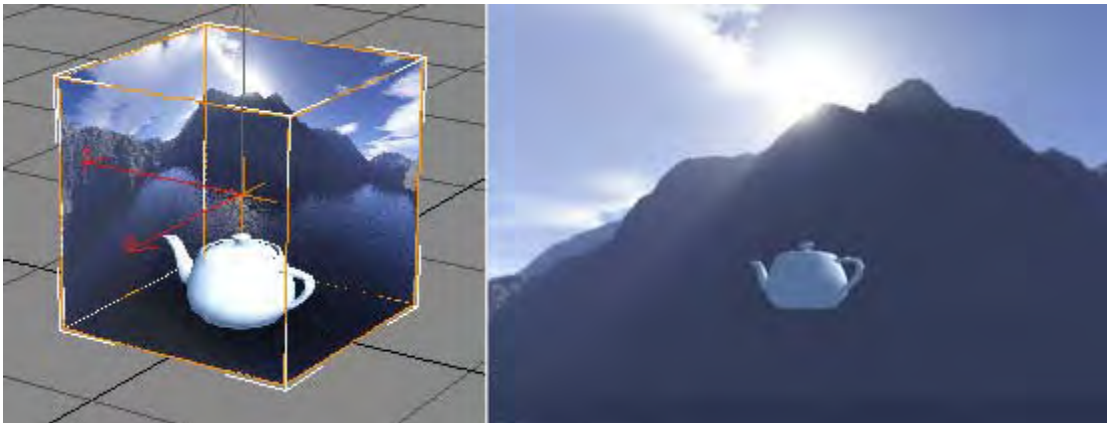


Fig. 6.9 Implementación de un *skybox*

6.3.3.2. Posprocesamiento

El posprocesamiento consiste en modificar la imagen resultante de dibujar todos los elementos de la escena, para crear una imagen final con diferentes efectos visuales. Por ejemplo, *ajuste de color* para incorporar de manera homogénea todos los elementos de la escena, el *desenfoque* que ayuda a dar a la imagen una apariencia más suave o un efecto más complejo como el *bloom*, lo cual genera un suave resplandor en los elementos dibujados.

Aunque algunos de los efectos de posprocesamiento son altamente exigentes en los recursos del GPU, en nuestro caso, se aplica un efecto de *bloom* con el

objetivo de dar una apariencia más profesional y terminada a nuestro videojuego. Este método es aplicado por medio de un componente en XNA lo que facilita su implementación dejando sin modificación alguna nuestra arquitectura.

6.4. Modelos Tridimensionales y Animación

El videojuego propuesto requiere de la implementación de un entorno virtual en tres dimensiones, lo que hace necesario el modelado y abstracción de cada uno de los objetos que lo componen.

Estos objetos pueden clasificarse en dos grupos: estáticos y animados. La diferencia principal entre ellos es que los objetos estáticos representan objetos inanimados dentro del entorno, por ejemplo: edificios o cajas; mientras que aquellos en el grupo de objetos animados se refiere principalmente a la representación de nuestros agentes, los cuales requieren de un conjunto de animaciones predefinidas que les permitirán un rango de movimiento simulando aspectos humano como lo es caminar o correr.

6.4.1. Modelos Tridimensionales estáticos

Para facilitar el manejo de un modelo tridimensional, XNA ofrece la clase *Model* la cual contiene toda la información necesaria, incluyendo una instancia de la clase *BasicEffect*, la cual define la forma final en como será dibujado el objeto en pantalla.

En nuestro proyecto, clases como *StaticWorldObject* y *WorldItem* contienen un parámetro del tipo *Model*, lo que les permite almacenar la información referente al modelo asignado a cada instancia de estas. Así en la inicialización del sistema cada uno es asignado con un modelo tridimensional previamente creado en un programa de autoría.

Para este sistema se crearon cada uno los objetos estáticos que se muestran en pantalla incluyendo terreno, edificios, señales, cajas, etc. Estos fueron modelados y texturizados usando la herramienta *Softimage|XSI Mod Tool* [SXMT09], la cual permite la exportación de los modelos en formato FBX, aceptado por XNA desde sus inicios.

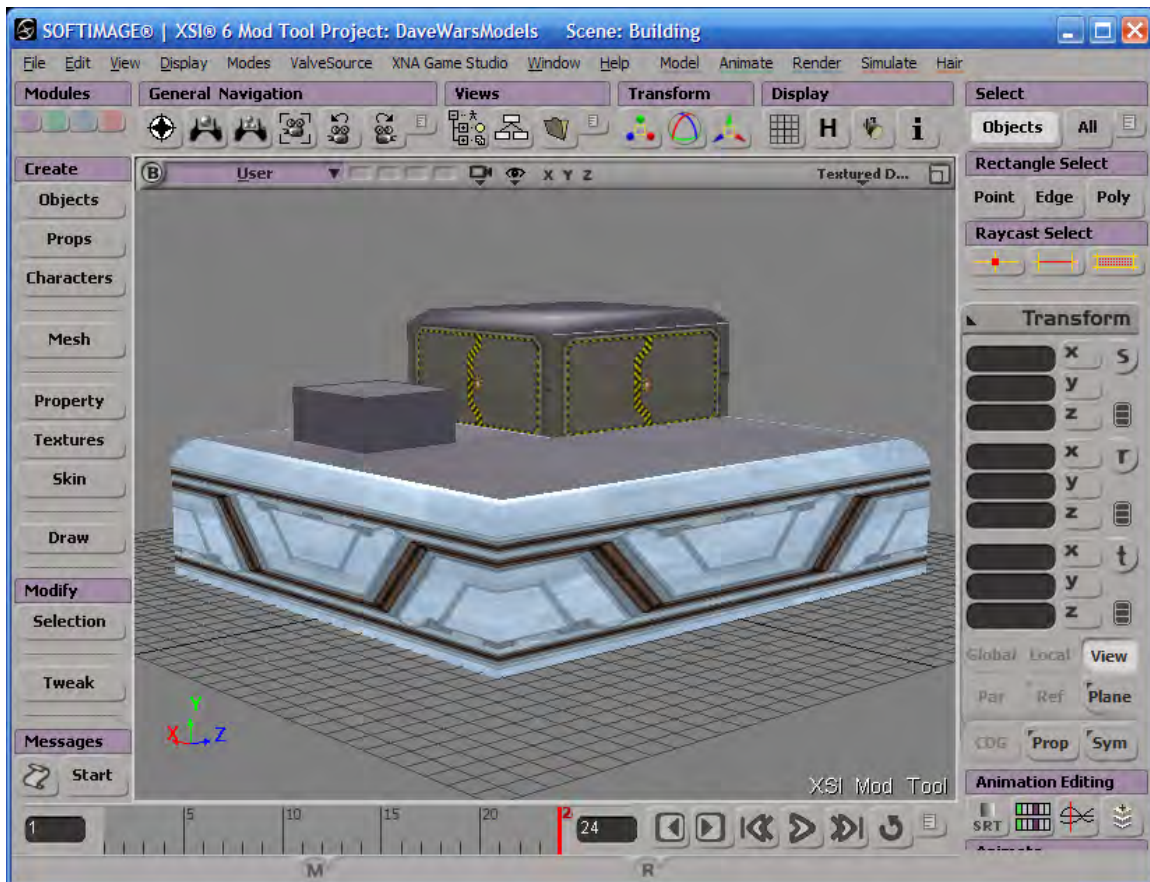


Fig. 6.10 Modelado de objetos con Softimage|XSI Mod Tool

6.4.2. Modelos de Agentes

El uso de personajes humanoides trae un gran problema a nuestra implementación. Evidentemente esperamos que cada uno de nuestros personajes se mueva y reaccione de una forma similar a como lo haría una persona, esto ayuda a formar una ilusión de que el personaje no es muy diferente a un humano. Sin embargo, cuando un modelo o su animación no tienen la calidad esperada estos pierden esa sensación de vida y rompen con la ilusión de comportamientos realistas que nuestro sistema busca.

El modelado y la animación de personajes son procesos complicados que llevan un largo tiempo en su desarrollo, lo que para propósitos de este proyecto resulta imposible crear en el tiempo establecido y para la cantidad de personajes que se requieren. Lo que hace entonces necesario una búsqueda de estos recursos y de una forma simple de implementarlos en nuestro sistema.

Uno de los aspectos que debemos considerar es que requerimos de modelos y animaciones predeterminadas que cumplan con nuestras necesidades bajo el contexto del videojuego propuesto. Es decir, necesitamos de personajes con

independencia en piernas y torso, además de un conjunto de animaciones específicas como es: caminar, disparar, morir, etc. Es aquí donde hemos escogido el uso del formato *Md3*, formato nativo del motor *Id Tech 3* creado por la empresa *Id Software* y usado en juegos comerciales como *Quake III Arena*.

Un archivo *Pk3* es un formato contenedor de los diferentes archivos digitales que conforman un modelo. Básicamente, encontramos tres archivos *md3* representando: la cabeza, el torso y las piernas del personaje. Adicionalmente se encuentran archivos con las texturas para cada malla además de una definición de las animaciones presentes. Generalmente, un modelo *md3* bien diseñado contiene el conjunto de animaciones enlistadas, que afectan de manera independiente ya sea a la malla del torso o de las piernas.

Animación	Piernas	Torso	Animación	Piernas	Torso
Gesto de Victoria		•	Nadar	•	
Disparar		•	Brincar	•	
Golpear		•	Caer	•	
Torso de pie		•	Brincar hacia atrás	•	
Recoger		•	De pie inactivo	•	
Levantarse		•	Agachado inactivo	•	
Caminar	•		Vuelta	•	
Caminar Agachado	•		Morir #1	•	•
Correr	•		Morir #2	•	•
Caminar hacia atrás	•		Morir #3	•	•

Tabla. 6.1 Animaciones contenidas en un archivo *md3*

Otra de las ventajas de escoger este formato, es que en la red se pueden encontrar innumerables modelos disponibles a la comunidad en forma gratuita, lo que nos permite seleccionar algunos de ellos para nuestros propósitos.

Aunque oficialmente la empresa creadora de este formato nunca ha ofrecido soporte para su uso, encontramos una gran cantidad de herramientas desarrolladas para la manipulación de estos archivos. *XNA Pk3Loader* [LOPR08], es un módulo implementado para la plataforma XNA que nos permite cargar, manipular las animaciones de este formato de forma sencilla y práctica. En la Figura 6.11 se aprecian dos de estos modelos, los cuales basan su aspecto visual en la clase *BasicEffect*. Lo anterior permite mantener un dibujo uniforme en conjunto con el resto de los objetos en escena.



Fig. 6.11 Modelos Md3 en XNA

Nuestro proyecto incluye la clase *Character*, la cual además permite heredar de *MovingEntity* para incorporar aspectos de movimiento en el entorno. Esta clase incluye un objeto de la clase *CPK3Model* el cual abstrae toda la funcionalidad del formato *md3*. De esta forma, por medio del método *SetAnimation* nuestra arquitectura finalmente tiene el soporte para representar personajes humanoides.

6.5. Movimiento autónomo

Hasta este momento hemos descrito la forma en como nuestro sistema implementa la visualización de objetos y personajes, y es en este punto donde ahora podemos incluir el *primer subsistema representante del motor de inteligencia artificial*.

Siendo una de las manifestaciones básicas de la autonomía de nuestros personajes, éstos deben contar con la facultad de desplazarse de un punto a otro, por lo que describimos las técnicas utilizadas para nuestro videojuego. El movimiento autónomo es uno de los elementos más importantes en este módulo, sin embargo, también es uno de los más difíciles de lograr pues cualquier deficiencia puede producir una sensación de poca inteligencia, ya que

uno de los primeras cosas que el jugador puede percibir es si los personajes se mueven de manera adecuada a la situación.

6.5.1. Comportamientos dirigidos

Inicialmente, proporcionamos a nuestros agentes la capacidad más elemental con el fin de modificar su posición actual; para alcanzar una posición objetivo. Para que un personaje pueda iniciar su movimiento este debe hacer uso de la clase *SteeringBehaviors* con el fin de activar alguno de los comportamientos descritos a continuación.

6.5.1.1. Buscar

El comportamiento *Buscar* o *Seek*, regresa una fuerza que dirige al agente directamente hacia una posición objetivo. Lo primero es calcular la velocidad deseada que indica la velocidad que necesita el agente para alcanzar el objetivo. Esta debe ser escalada a la velocidad máxima especificada para el agente.

La fuerza de dirección es aquella que sumada con la velocidad actual del agente permite obtener la velocidad deseada. Esto se observa mejor en la Figura 6.12.

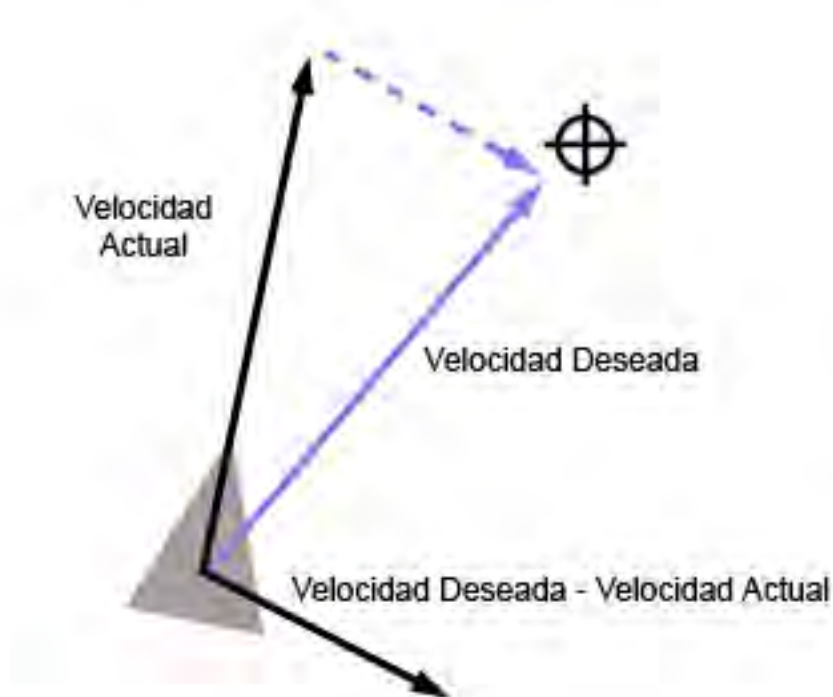


Fig. 6.12 Cálculo de vectores para el comportamiento buscar

Este comportamiento es muy útil para implementar otros aún más complejos y en nuestro caso será utilizado para dirigir a los personajes directamente a una posición deseada.

6.5.1.2. Huir

El comportamiento *Huir* o *Flee* es lo opuesto a *Buscar*. En lugar de producir una fuerza guía hacia el objetivo, *Huir* produce una fuerza en sentido contrario, lo que fuerza al agente a alejarse de la posición indicada.

Este comportamiento puede ser ajustado para generar la fuerza de dirección únicamente cuando se encuentre dentro de una distancia mínima, lo que evidentemente le agrega un grado más de credibilidad que nos permite que los agentes que usen este comportamiento se alejen de su enemigo, sólo cuando este se encuentre a una distancia cercana.

6.5.1.3. Llegar

Aunque *Buscar* nos permite dirigir a un agente hacia la dirección adecuada, a menudo, es necesario que un personaje se mueva hacia un punto y se detenga de manera suave. *Llegar* o *Arrive* genera una fuerza de dirección con estas características, donde su magnitud disminuye de manera proporcional conforme la distancia entre el agente y su objetivo se hace más pequeña. Cuando la distancia al objetivo es igual a cero entonces la fuerza de dirección es un vector de longitud cero.

6.5.2. Planeación de rutas

La planeación de rutas es un concepto que agrega un grado más a nuestra percepción del movimiento autónomo inteligente. Cuando una persona toma su automóvil para trasladarse de su casa al trabajo tiene una imagen mental del camino que seguirá, el cual generalmente procura ser el más eficiente para ahorrar tiempo y gasolina. De manera similar nuestros agentes utilizan esta idea analizando el mejor camino que pueden seguir para llegar a su destino.

6.5.2.1. Mapas de navegación

Al igual que un automovilista, nuestro agente requiere de un mapa del entorno que le permita conocer los caminos posibles que pueda tomar además de alguna heurística que pueda utilizar para determinar si un camino es más costoso que otro. Este mapa virtual es llamado *Grafo de Navegación*.

Buckland en su libro [BUCM05] nos ofrece una descripción de los tipos y problemas que ofrecen las distintas aproximaciones a los grafos de navegación, así como algunas técnicas para su creación. En nuestro caso, este es implementado a través de la clase *NavGraph* la cual incluye una lista de nodos, cada uno a su vez contiene una lista de aristas que salen de él hacia otros nodos. En la Figura 6.13 puede observarse una visualización del grafo de navegación utilizado por cada agente en el sistema.

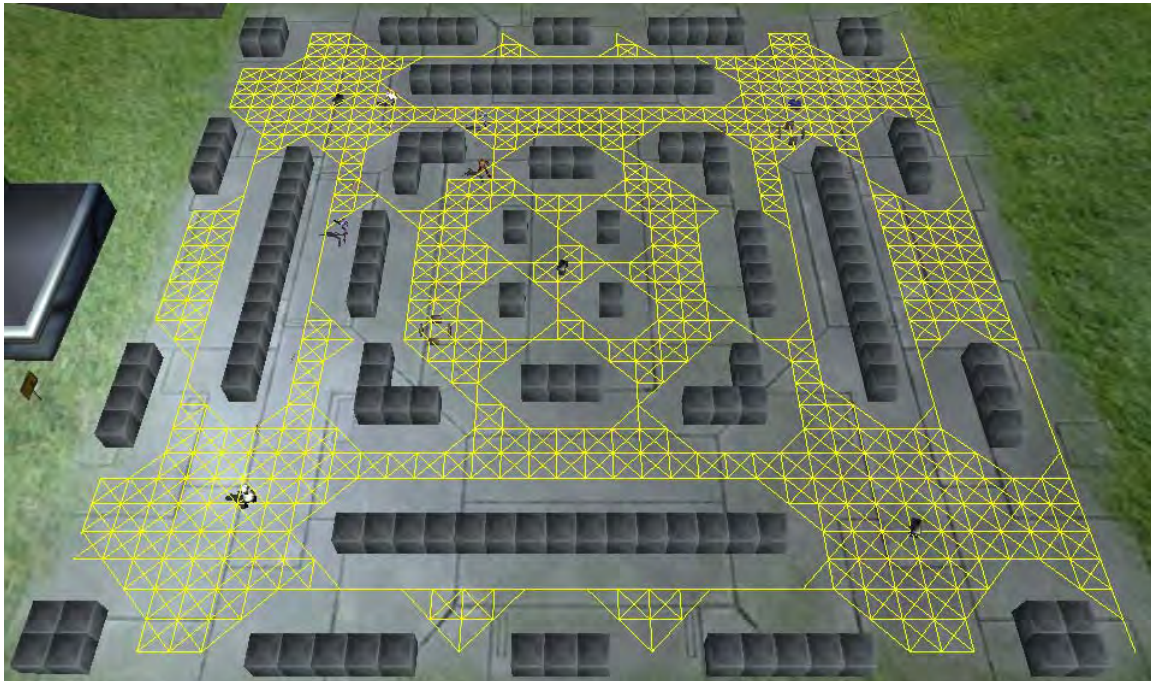


Fig. 6.13 Implementación del Grafo de Navegación

6.5.2.2. Funcionamiento del planificador de rutas

La funcionalidad para la planificación de rutas esta contenida en la clase *PathPlanner*. Para facilitar la creación de nuevos niveles, el grafo de navegación se genera al momento de cargar la información correspondiente al entorno, para posteriormente asignar una referencia a cada entidad que lo requiera.

Cada vez que un agente requiere trasladarse de un punto a otro, hace una petición a su sistema de navegación, en este caso su propia instancia de *PathPlanner*. Este procesa la posición origen y destino para obtener los nodos más cercanos a cada uno de estos. Con un nodo inicial y uno final es posible ahora utilizar el grafo de navegación para realizar la búsqueda. Existen distintos algoritmos para realizar búsquedas en grafos cada uno de ellos con ventajas y desventajas, en nuestro caso utilizaremos el *algoritmo de Dijkstra* [BUCM05], sin embargo, nuestra arquitectura cuenta con la flexibilidad de incorporar otras búsquedas y seleccionar aquella que se adapte más a la situación.

La implementación del algoritmo de Dijkstra se encuentra en la clase *GraphSearchDijkstra*, la que ofrece dos métodos útiles para el planeador de rutas: *Search* y *GetPathToTarget*. El método *Search* encargado de ejecutar la actual implementación del método de búsqueda recibe como parámetros el nodo inicial y el nodo destino, siendo generalmente aquí uno de los procesos con más consumo de recursos donde existen gran cantidad de técnicas para mejorar su rendimiento. El planeador de rutas al recibir una señal de que un camino se ha encontrado hacia el nodo destino, entonces solicita la lista de nodos que lo componen por medio del método *GetPathToTarget*. Finalmente, con esta lista es posible extraer de cada nodo su posición espacial, agregando al inicio la posición actual del agente y al final la posición destino. La relación de estas clases puede observarse en la Figura 6.14.

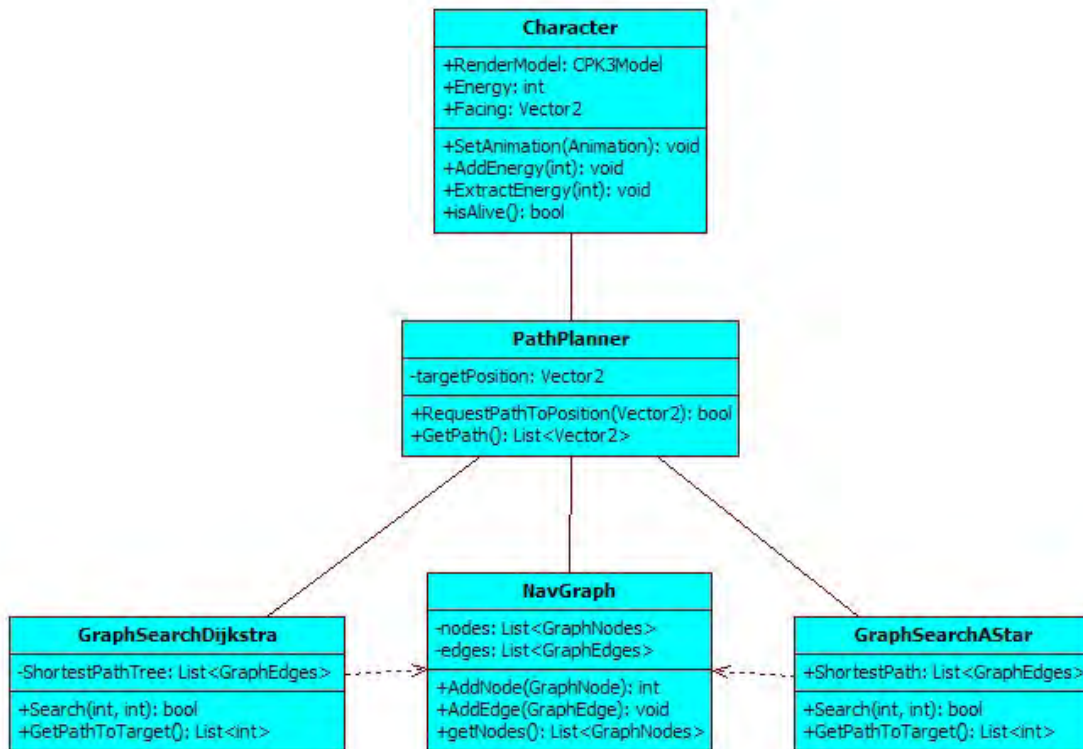


Fig. 6.14 Clases que componen al planeador de rutas

Ahora con una ruta disponible es responsabilidad de otro módulo guiar al agente a través de cada una de las posiciones que la componen. Esto ahora resulta una tarea sencilla si consideramos que los comportamientos dirigidos sirven para este propósito. Utilizando *Buscar* para cada posición que corresponde a la ruta a excepción del último donde ejecuta el comportamiento *Llegar* para terminar el recorrido de una manera más elegante y natural.

6.6. Modelo de metas

Con un entorno lleno de objetos, donde los personajes pueden moverse de manera autónoma y comunicarse con otras entidades, es necesario darles una motivación, una meta que los impulse a interactuar con todo el mundo que hemos creado para ellos.

Muchas técnicas se han planteado para dotar a los agentes de un sistema de razonamiento que les permita generar acciones que satisfagan las metas para las que fueron diseñados [TOZP02] [BURR01] [ORKJ06] [SCHB04] [THOA02] [OBRJ02]. Para nuestro caso de estudio proponemos el uso de un sistema de planificación que utiliza metas jerárquicas, ya que el análisis cognitivo que se realizó en el capítulo 5, nos permite contar con una descomposición del comportamiento en tareas simples y bien definidas. Además un sistema de este tipo nos ofrece los siguientes beneficios:

- *Adaptabilidad.* No importando la situación del juego, la IA generará metas para los agentes y éstos tratarán de seguirlas incluso en aquellas situaciones del entorno que no fueron consideradas.
- *Sensación más humana.* El hecho de que un agente tenga un plan para cumplir sus metas, genera una sensación hacia el usuario de que existe un proceso mental similar al humano y no solo un comportamiento reactivo hacia cada evento del entorno.
- *Facilidad para depurar.* Cualquier implementación de IA ofrece al programador una dificultad adicional para depuración y refinamiento. Afortunadamente, este esquema nos permite visualizar en todo momento lo que esta pensando cada agente, es decir, podemos obtener una imagen completa del árbol de metas que está persiguiendo en un instante dado y observar la submeta activa en el momento, así como el plan completo que lleva a cabo para cumplir su meta principal.
- *Escalabilidad.* La independencia de metas y la fragmentación de cada una en submetas, nos permite de una manera sencilla agregar nuevas metas para incrementar el rango de comportamientos. Esto sin alterar de manera significativa la arquitectura ya diseñada.

6.6.1. Composición de las metas

La plataforma que se desarrollo para este proyecto implementa un sistema de metas usando el patrón de composición, el cual agrupa una serie de objetos en estructuras tipo árbol para representar una jerarquía, permitiéndonos tratar a cada objeto de una manera uniforme no importando su complejidad.

Denotamos esta naturaleza jerárquica ejemplificándola. Supongamos que deseamos poseer una nueva computadora, ésta será nuestra meta principal. Sabemos que para adquirirla debemos dirigirnos a una tienda, hablar con el encargado y cerrar el trato, pero antes de eso debemos tener en nuestras manos el dinero suficiente para comprar la computadora que cumpla con nuestras necesidades, entonces una opción es tener el dinero en el banco. Debemos dirigirnos a él para obtenerlo, de lo contrario, tenemos que realizar un nuevo plan para obtener el tan necesario dinero para cumplir la meta de obtener una nueva computadora para probar nuestra implementación.

Como se puede observar en el ejemplo anterior, para satisfacer cada meta debemos descomponerla en submetas más pequeñas y a su vez tratar de descomponer estas aún más. Siguiendo este proceso finalmente obtendremos una meta tan simple que no requerirá de submetas más pequeñas y podrá realizarse por medio de una acción. Es aquí donde el patrón de composición nos será de utilidad.

La forma en como este patrón funciona aplicándolo a nuestro enfoque de metas es creando una clase abstracta, que representa tanto una meta compuesta como una atómica, es decir, una meta que puede ser cumplida por medio de una simple acción. De esta forma un agente podrá procesar sus metas sin importar lo complejo de éstas. Las ideas anteriores pueden observarse en la Figura 6.15.

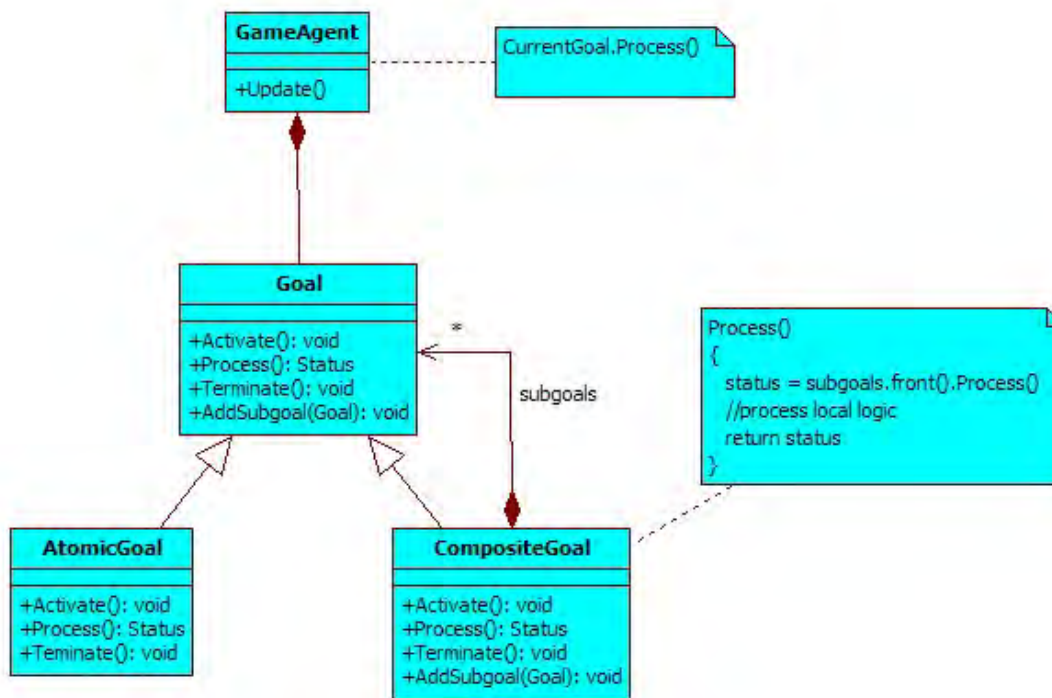


Fig. 6.15 Patrón de Composición aplicada a las metas del agente.

Las clases *Goal* y *CompositeGoal* serán nuestros modelos sobre los cuales cada meta será creada. Si una meta requiere de otras para ser completada entonces ésta heredará sus propiedades de *CompositeGoal*, que le proporcionará una estructura tipo pila para almacenar todas las submetas que la compongan. De otra manera, si la meta es suficientemente simple para no requerir de otras entonces será implementada como una meta atómica, que al cumplir con su lógica interna se considerará completada y será retirada de la lista de metas de su meta superior.

Podemos notar que el cliente de este patrón, en nuestro caso cada uno de los agentes en el entorno, procesará su meta principal lo que ocasionará que toda la estructura sea actualizada de igual manera. Permitiendo que en cada caso pueda cumplirse una meta a la vez hasta alcanzar la meta primaria.

6.6.2. Jerarquía de metas para los agentes

En el capítulo 5, se diseñaron los comportamientos que nuestros agentes deberán seguir a lo largo del juego, esto nos permitió establecer tres metas principales que deben ser implementadas: *sobrevivir*, *proteger al equipo* y *atacar al enemigo*. Cada agente deberá estar persiguiendo activamente una de estas metas.

La meta *sobrevivir* esta implementada por medio de la clase *GoalSurvive* la que contiene el siguiente conjunto de clases y que representa sus submetas:

- *GoalGetItem*. Indica al agente que debe encontrar y recolectar un ítem del tipo indicado. Esto lo logra: 1) identificando el lugar conocido más cercano donde se encuentra este, 2) desarrollar un plan para llegar hasta él, y, 3) si el ítem aún se encuentra ahí recolectarlo enviando los mensajes correspondientes.
- *GoalTakeCover*. En una situación donde un enemigo se encuentra presente, esta meta le permite al agente encontrar una cubierta cercana a su posición y permanecer ahí para recibir el menor daño posible.
- *GoalEscape*. Simplemente le indica al agente que debe alejarse del enemigo hasta estar a una distancia considerada segura. Esta es una meta atómica pues sólo es necesario activar el comportamiento dirigido *Huir* sin necesidad de planear una ruta de escape más elaborada.

La meta *proteger al equipo* esta construida en la clase *GoalProtectTeam* y esta a su vez esta formada de las siguientes submetas:

- *GoalReinforcements*. Indica que su compañero ha solicitado ayuda y es necesario trasladarse a la posición donde este se encuentre. Con el fin de atacar a los enemigos cercanos.
- *GoalHealPartner*. Indica que su compañero ha sido derrotado por lo que no puede moverse, así que debe trasladarse a su posición y reanimarlo.
- *GoalGetItem*. Al igual que en el caso de la meta de sobrevivir, indica que debe recolectar el ítem requerido. El uso de esta submeta ofrece una aproximación distinta para proteger al equipo, pues en caso de no existir ninguna solicitud de ayuda el personaje puede recolectar ítems para incrementar el estado general del equipo, aumentando sus probabilidades de supervivencia.

La última meta que nuestros agentes pueden perseguir es *atacar enemigo* encapsulada en la clase *GoalAttackTarget* y llevada a cabo por las submetas:

- *GoalHuntTarget*. En caso de que ningún enemigo este en el rango de visión del agente, esta submeta le indicará seguir varios caminos en busca de un enemigo al cual pueda atacar.
- *GoalTakeCover*. En caso de que exista un enemigo presente y una cubierta cercana a nuestra posición se trazará un plan, para utilizar la cubierta e incrementar las posibilidades de derrotar al enemigo.
- *GoalDodgeSideToSide*. Si un enemigo esta presente y no hay forma de atacar de forma segura, la mejor opción es esquivar los disparos, así esta meta mueve al agente de un lado a otro convirtiéndolo en un blanco más difícil.

Algunas metas existen de manera recurrente en otras de mayor jerarquía, por ejemplo, cuando se requiere de un ítem o el agente intenta ayudar o reanimar a su compañero, se requiere de trasladarse a una posición específica, lo que es producido por las siguientes metas:

- *GoalMoveToPosition*. Cuando el agente decide que debe trasladarse de un punto a otro, esta meta es agregada a su sistema solicitando la búsqueda de un camino adecuado por medio del planificador de rutas.
- *GoalFollowPath*. Cuando el resultado de la búsqueda de un camino es positivo, entonces es posible solicitar la posición de cada punto que lo conforma. Esta meta se encargará de seguir en el orden requerido por cada uno de los puntos hasta llegar al último de ellos.

- *GoalSeekToPosition*. En el nivel inferior el seguimiento de un camino requiere trasladar al agente de una posición a otra. Esta meta realiza precisamente esta acción guiando al personaje de su posición actual al punto que se especifica, utilizando el comportamiento dirigido *Buscar*.

Finalmente, se necesita de una forma de concentrar todas estas metas y donde puedan evaluarse los distintos aspectos que permiten determinar cuál de las metas es la de mayor importancia, y por tanto, la que debe ser colocada en la parte superior de la jerarquía. La clase *GoalThink* es la responsable de este proceso. Por medio de su método *Arbitrate* se verificarán estas condiciones agregando la meta requerida en caso de que esta no se encuentre ya en la pila. Posteriormente, cada actualización del agente llamará al método *Process* lo que procesará la meta actual ya asignada y todas sus submetas creadas al momento.

6.7. Modelo Emocional

Ahora en este punto contamos por fin con un videojuego completo, que incluye un entorno en donde un grupo de agentes interactúan entre si siguiendo sus propias metas. Éstas no se detendrán hasta verse cumplidas, a excepción de que algún cambio en el entorno o en si mismos les impida lograrlo.

Nuestra hipótesis inicial pretende que, a partir de un videojuego diseñado con las técnicas conocidas y comúnmente utilizadas, sea posible incrementar la sensación de realismo en el comportamiento de los personajes. Lo anterior al aplicar un modelo emotivo que afecte al proceso de toma de decisiones.

Para lograrlo, incluimos un par de elementos extras que nos ayudarán a crear las emociones sintéticas que influirán en nuestro agente experimental. Estos elementos están contruidos con base en el modelo teórico investigado a lo largo de este proyecto y no afectan de manera significativa la arquitectura del sistema lo que asegura su exitosa implementación.

6.7.1. Generador de Acontecimientos

El *generador de acontecimientos* descrito en la *arquitectura emotiva* expuesta en el capítulo 4, es parte importante del modelo teórico pues nos ayuda a identificar cuales son los aspectos del mundo virtual que deben ser monitorizados en busca de cambios para ser interpretados por los agentes. Por ejemplo un acontecimiento relevante, es decir, aquellas variables consideradas en el diseño de los personajes y sus emociones.

La implementación corre a cargo de la clase *EventGenerator* que como muchos otros objetos hereda de *CoreEntity* e implementa la interface *IMessageReceiver*. Estas características le permiten tanto enviar como recibir mensajes,. El envío de mensajes es de utilidad para que otras entidades comuniquen ciertos eventos que pueden evolucionar en acontecimientos, es decir, aquellos descritos por la *teoría OCC*.

El *generador de acontecimientos* actualiza su estado en cada ciclo en búsqueda de información relevante. Esto junto con los datos recibidos por otras entidades le permite identificar un acontecimiento. El cual encapsula en un mensaje del tipo *Msg_Emotion_ProcessEvent* el tipo de acontecimiento que representa, posteriormente, será enviado a todas las entidades registradas en su lista de entidades emotivas. Cada entidad que reciba este mensaje será responsable de interpretarlo y reorganizar su estado de manera coherente.

6.7.2. Entidad emocional

Para que una entidad pueda ser capaz de tener emociones, ésta debe tener un modelo de metas. Este se encuentra implementado y es descrito en este mismo capítulo. Además debe ser capaz de identificar los acontecimientos que acontecen en el entorno. Todo esto es llevado a cabo por el generador de acontecimientos y transmitido por medio del modelo de mensajes ya incorporado al sistema.

Para obtener entonces una entidad emotiva ahora sólo es necesario que cada agente pueda interpretar los acontecimientos recibidos y afectar su estado, pero para lograrlo es necesario que describamos la implementación final que posee cada uno de los agentes.

6.7.2.1. La clase soldado

La clase *Soldier* es tal vez una de las más importantes en este proyecto. Aquí se encuentran organizados cada uno de los módulos que nos permite observar a los agentes moverse en su entorno e interactuar con él. La Figura 6.16 muestra como se encuentra estructurada.

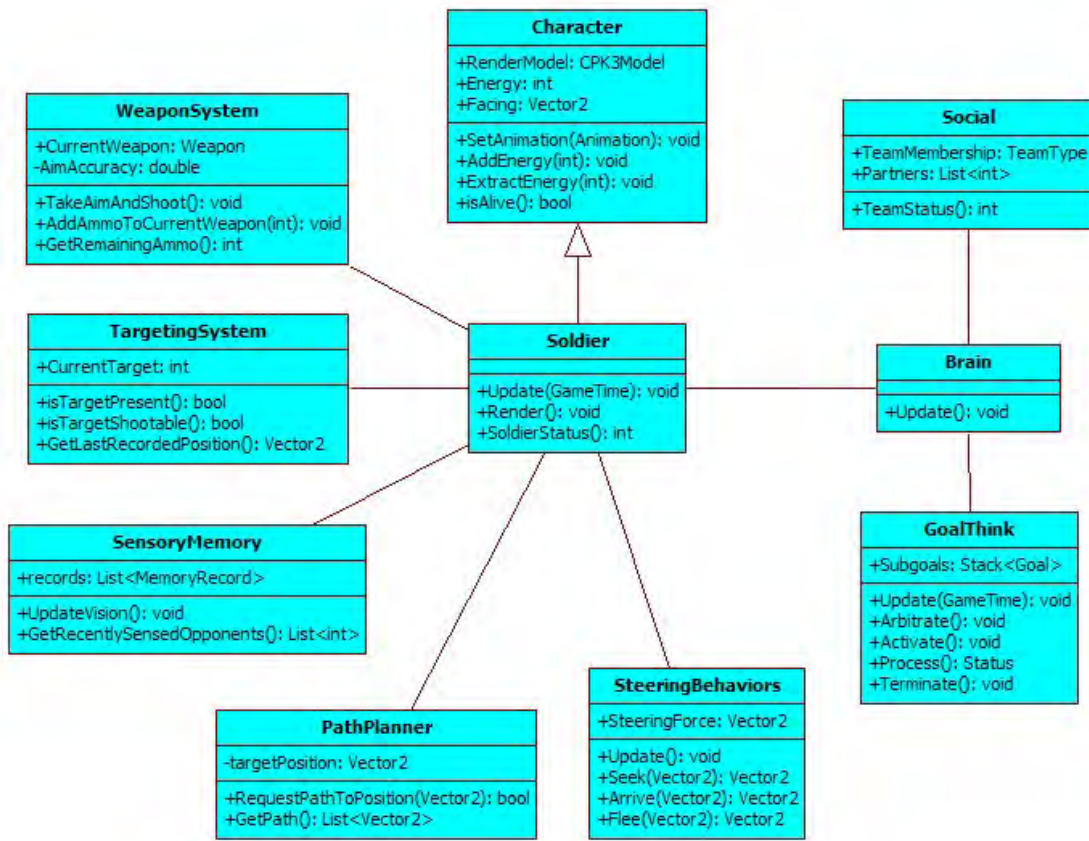


Fig. 6.16 Organización interna de los agentes.

Algunos de estos subsistemas se han descrito previamente a lo largo de este trabajo, sin embargo, se muestra una descripción breve del funcionamiento de cada uno de estos como parte del agente:

- *SteeringBehaviors*. Define la parte de locomoción del agente que le permite moverse en el entorno.
- *PathPlanner*. Similar a un sistema de navegación comercial este le permite al agente proporcionar su posición y encontrar el mejor camino posible para llegar a su destino.
- *SensoryMemory*. Agrupando el sentido de visión y una memoria a corto plazo, este módulo ayuda a identificar a los enemigos en el rango de visión así como recordar la presencia de cada uno de estos en caso de que no puedan ser vistos por la oclusión de algún otro objeto.
- *TargetingSystem*. Usando el sistema de visión y memoria permite seleccionar un enemigo como objetivo para atacar de acuerdo a un criterio específico.

- *WeaponSystem*. Ayuda al agente a cambiar su orientación y disparar su arma en caso de que el sistema que selecciona el objetivo haya captado a un enemigo al que se le pueda disparar.
- *Brain*. Este componente agrupa algunos aspectos que consideramos como aspectos más cognitivos. Principalmente, esta formado por el sistema deliberativo que a su vez esta constituido por el sistema de metas jerárquicas y un bloque con información relativa a la sociedad de agentes. Por ejemplo, ¿a qué grupo pertenece el agente? o ¿quienes son sus enemigos?

6.7.2.2. Cerebro emocional

Ya que conocemos la estructura interna de nuestros agentes, podemos identificar el punto exacto donde un sistema emocional puede influir. Es decir, si nuestra propuesta indica que modificaremos la toma de decisiones, entonces el mejor lugar para ubicarlo es el subsistema *Brain*.

Según la arquitectura emotiva propuesta, cada individuo debe contener un *modulo afectivo individual (MAI)* responsable del manejo de las emociones y *Brain* es el mejor lugar para ubicar este componente. La clase *EmotionalBrain* contiene esta funcionalidad permitiendo al agente procesar los mensajes relacionados con las emociones e interpretarlos.

La interpretación de cada mensaje que abstrae el concepto de acontecimiento permite al *MAI* manipular las intensidades de cada una de las emociones que pueda contener. En el caso de estudio, se implementaron dos emociones por medio de la clase *Emotion* la cual incluye parámetros como son: su tipo, intensidad, umbral y factor de decremento. Con cada estímulo hacia las emociones, éstas son analizadas para determinar si han superado su umbral y entonces determinar si la emoción se presenta como activa.

Ya que el agente ahora puede determinar si esta experimentando una emoción y el tipo de ésta, es responsabilidad del sistema de razonamiento darle una utilidad a este suceso. La clase *EmotionalGoalThink* hace uso de toda la arquitectura de metas ya descrita, sin embargo, sobrescribe la funcionalidad del método *Arbitrate* que teniendo el conocimiento de la emoción activa permite cambiar la jerarquía de las metas justo como se describió en el diseño del modelo emocional. De esta forma, un agente que determine a través de su propio módulo afectivo individual que esta sintiendo desprecio hacia su compañero, reorganizará sus metas para dar prioridad a su supervivencia y ataque enemigo. Lo anterior antes que la protección del equipo. La relación de todas estas clases se puede apreciar en la Figura 6.17.

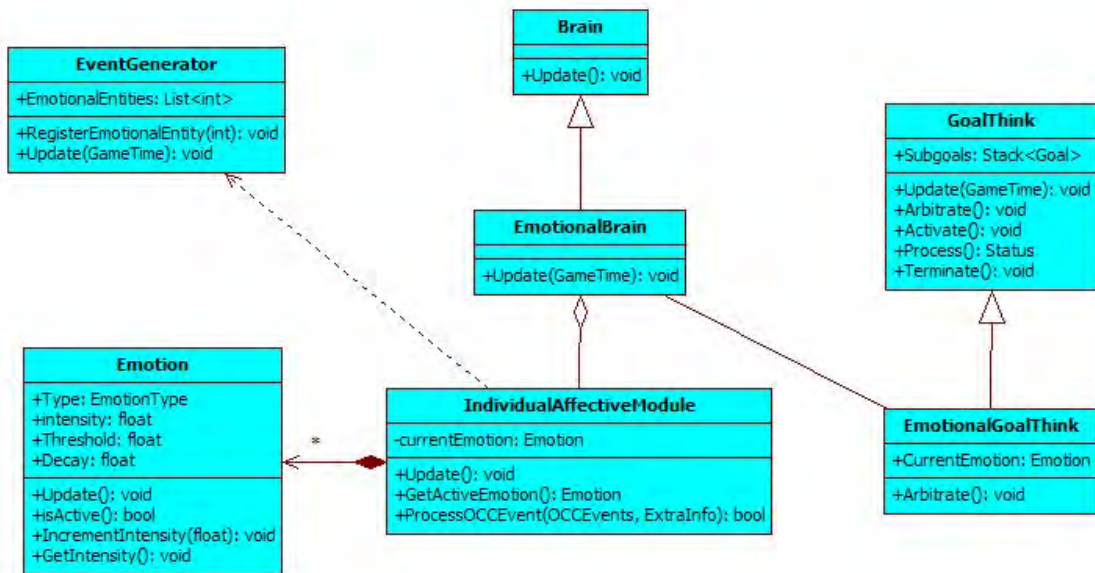


Fig. 6.17 Cerebro Emocional



Capítulo 7

Presentando: *Dave Wars*

Se desarrollo e implemento un videojuego con el modelo emotivo diseñado. En este capítulo se presenta el sistema titulado *Dave Wars*. Una introducción conceptual de este juego puede encontrarse en el apéndice A. A continuación se describen las pantallas que lo constituyen. Mostraremos la interfaz y el control que cada jugador debe conocer, para finalmente describir algunos casos de la presencia del modelo emotivo y su influencia en el resultado final.

7.1. Descripción de Pantallas

7.1.1. Menú Principal

Esta es la primera pantalla que cualquier jugador obtendrá al iniciar el sistema. Mostrando el título del videojuego, incluye además el menú principal sobre el cual el jugador podrá realizar las siguientes acciones:

- *Play Game*. Esta opción es la que inicia el juego mismo, permitiendo al usuario controlar su personaje, interactuando en el entorno con los otros agentes cumpliendo sus objetivos hasta que las condiciones necesarias de término del juego se cumplan.
- *Help*. Presenta una pantalla de ayuda con información para el control del agente.

- *Credits*. Muestra una pantalla con los créditos de las personas que colaboraron con la creación de este videojuego.
- *Exit*. Esta opción permite salir de la aplicación.

Esta pantalla permanecerá activa mientras el usuario selecciona un elemento del menú y presione el botón A para activarlo.

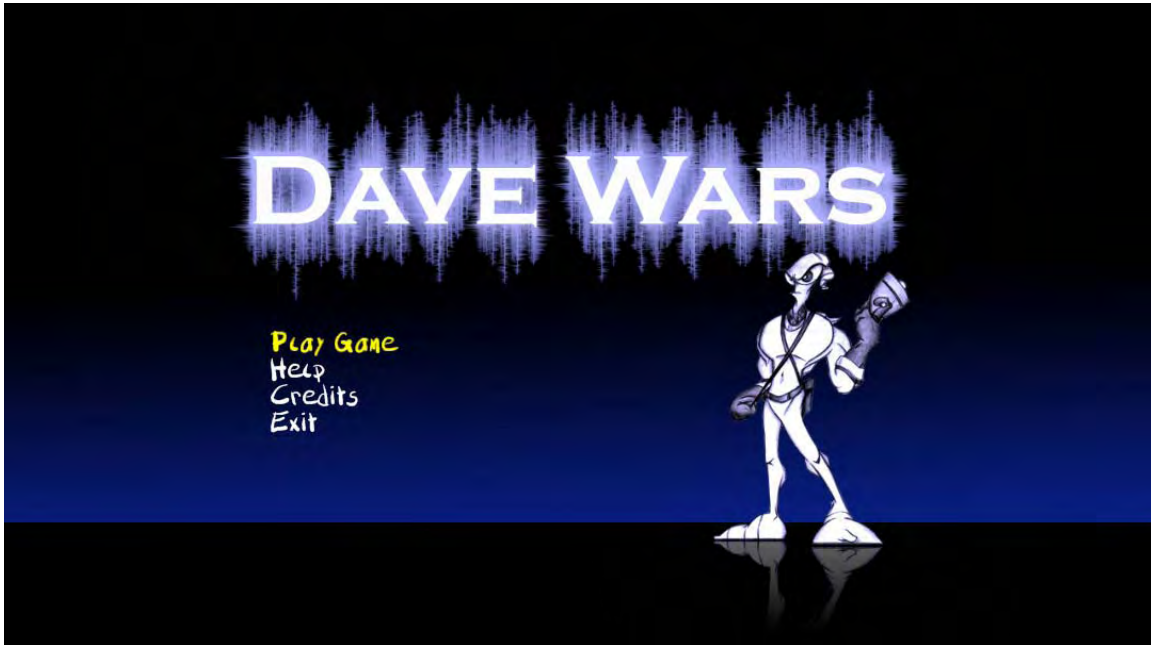


Fig. 7.1 Menú Principal

7.1.2. Pantalla de Ayuda

Seleccionando la opción *Help* del menú principal, se muestra una imagen que presenta al usuario una vista esquemática del control de su personaje por medio del *gamepad* del Xbox 360.

Una explicación más profunda de este control es expuesto en la sección 2 de este capítulo.



Fig. 7.2 Pantalla de Ayuda

Con nada más que hacer en esta pantalla el usuario puede regresar al menú principal presionando el botón B, como lo muestra la indicación en la esquina inferior derecha de la Figura 7.2.

7.1.3. Elementos del juego

Al iniciar un juego se presentan diversos elementos los cuales forman de hecho nuestro mundo virtual. Su identificación es clave para comprender la dinámica que se genera con cada objeto. Las Figuras 7.3 y 7.4 muestran como son vistos en pantalla los elementos descritos:

- *Avatar usuario.* Este es el personaje principal del juego del cual tendrá control el usuario.
- *Compañero.* Este personaje se considera el compañero del jugador por encontrarse en el mismo equipo y compartiendo las mismas metas. Es también el sujeto de prueba de nuestro experimento emocional.
- *Enemigo.* Existiendo cuatro diferentes, los enemigos, son los personajes restantes que forman el segundo equipo en nuestro entorno. Estos comparten la meta de eliminar tanto al jugador como a su compañero y aunque comparten las mismas capacidades que el jugador, al ser derrotados revivirán después de un periodo de tiempo para reiniciar su ataque.

- *Ítem.* Cada personaje cuenta con su propia cantidad de energía y municiones. Para recuperar cada uno de estos, un personaje debe recolectar un ítem del tipo adecuado.
- *Proyectil.* Dado que nuestros personajes pueden atacar a otros, la acción de disparar su arma genera un proyectil que viajará en línea recta hasta colisionar con algún otro objeto en el entorno.
- *Cubierta.* Como un elemento para dar más dinámica a los ataques de los personajes, estos pueden colocarse al centro de este elemento ofreciéndoles un decremento en el daño producido por los proyectiles recibidos. El uso continuo de una cubierta ocasionará que esta se degrade desapareciendo de la vista al igual que sus efectos, sin embargo, esta reaparecerá al término de un tiempo específico.
- *Estado del personaje.* Este elemento nos permite conocer el estado actual de las distintas variables relacionadas con los personajes en el equipo del jugador.
- *Metas activas.* Como parte de nuestro proyecto esta información es mostrada en pantalla, para permitir la observación de las distintas metas que el compañero del jugador esta persiguiendo en cada momento.



Fig. 7.3 Interfaz del juego

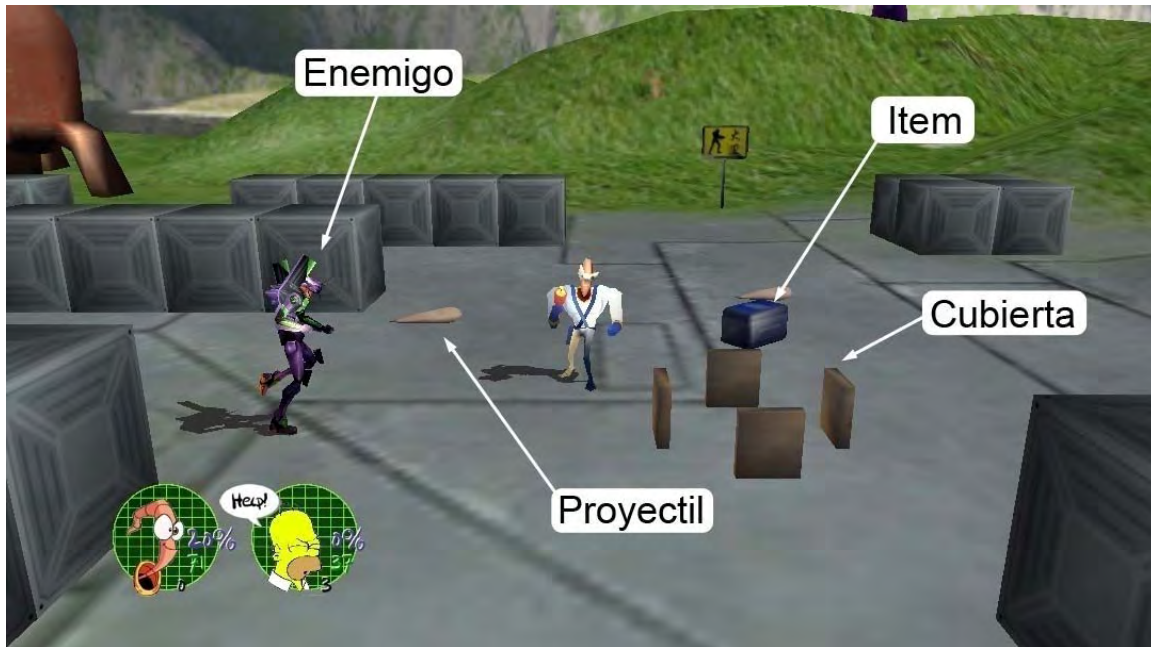


Fig. 7.4 Interfaz del juego

7.1.4. Detalle de la interfaz del juego

La interfaz de juego nos presenta información más detallada sobre el estado interno de los personajes tanto del jugador como de su compañero. Los elementos básicos se muestran en la Figura 7.5 y son descritos a continuación:

- *Energía*. Representa la variable que da vida al personaje, cuando esta llega a cero debido a los ataques enemigos entonces el personaje cae derrotado y no puede realizar ninguna actividad a excepción de solicitar ayuda a su compañero. Cuando la energía tanto del jugador como su compañero llegan a cero es la causa suficiente y necesaria para declarar el término del otro.
- *Munición*. Corresponde a la cantidad de disparos disponibles que tiene el personaje para usar su arma.
- *Enemigos Eliminados*. La puntuación de los personajes que han alcanzado a lo largo del juego, directamente relacionada con la cantidad de enemigos eliminados.

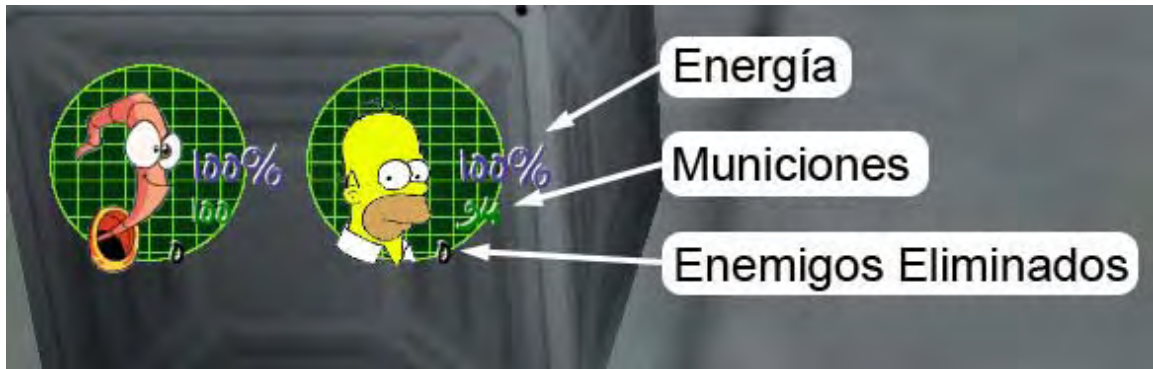


Fig. 7.5 Propiedades de la representación del personaje

Otros elementos mostrados en la Figura 7.6 tienen un significado más relativo a la conducta de los agentes. La *señal de ayuda* es una forma de comunicación que informa a sus compañeros de equipo cuando un personaje solicita refuerzos, en caso de encontrarse aún de pie o cuando solicita curación debido a que ha sido derrotado. Esta señal desaparece una vez que estas necesidades han sido satisfechas.

El último elemento de esta interfaz corresponde a la *expresión de su estado*, es decir, una representación de su organización interna. Aunque este trabajo no tiene relación con aquellos cuya investigación se centra en la expresión de las emociones, este aspecto nos permite tener una forma sencilla y clara para determinar la emoción activa de nuestro personaje. Dado el presente diseño, contamos con dos emociones sumadas a un estado afectivo neutro y al estado de inactividad del personaje. Lo anterior ofrece un rango de cuatro expresiones que pueden mostrarse disyuntivamente a lo largo del juego.

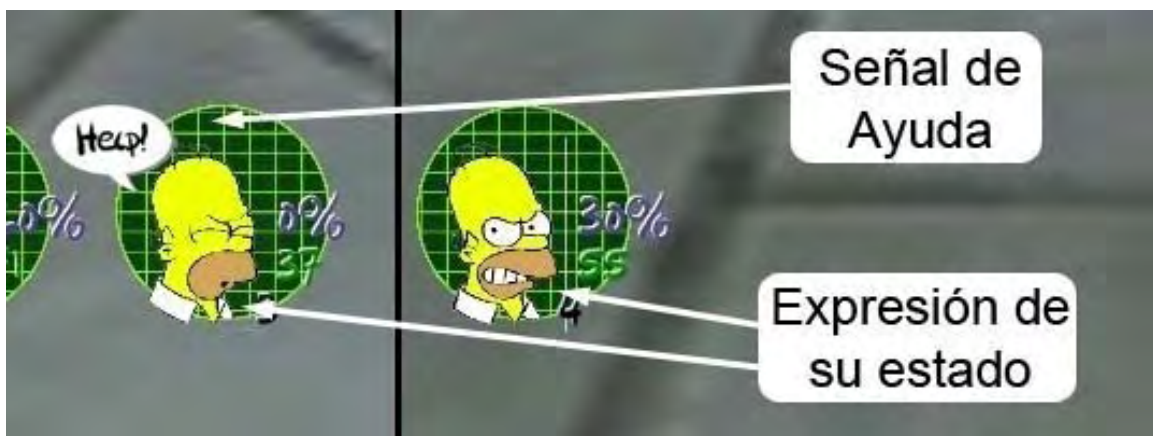


Fig. 7.6 Estado del personaje

7.1.5. Pantalla de término

La última pantalla creada para este proyecto se presenta después de que las condiciones de término son alcanzadas. Mostrando al usuario que el juego actual ha concluido; le presenta la puntuación alcanzada y le permite regresar al menú principal con las opciones ya antes mencionadas.



Fig. 7.7 Pantalla de término

7.2. Control de Usuario

7.2.1. Control de Xbox 360

Todo el sistema del control diseñado para nuestro videojuego es realizado a través del control usado en la consola Xbox 360 o *gamepad*. Este fue creado pensando en las necesidades de navegación de los entornos tridimensionales de los videojuegos actuales, además ofrece una compatibilidad práctica de conexión a una computadora personal con puerto USB y el sistema operativo de Microsoft.

La Figura 7.8 muestra una imagen del *gamepad* indicando cada una de las partes que lo componen.



Fig. 7.8 Control de Xbox 360

La decisión de diseño de incorporar este dispositivo de entrada radica tanto en su facilidad de uso como en su familiaridad por parte de los jugadores. Además el soporte ofrecido por la plataforma XNA nos permite su uso uniforme en cualquiera de las plataformas sobre las que puede ejecutarse el sistema.

7.2.2. Control del jugador

El juego presenta una pantalla de ayuda que ofrece información relacionada con el *gamepad* y las acciones que realiza el agente en el entorno. A continuación describimos a detalle este mapeo que sólo afecta al avatar del usuario.

- *Left stick*. Permite el movimiento del agente sobre un plano bidimensional que representa el suelo del entorno sobre el cual se mueven los personajes.
- *Right stick*. Cambia la orientación del personaje en la dirección indicada.
- *Left trigger*. Permite realizar un ataque cercano, es decir, un ataque de corto alcance que no requiere de municiones, lo que es de utilidad en caso de que el usuario agote sus municiones.
- *Right trigger*. Ejecuta la forma principal de ataque por parte de los agentes, disparando su arma restando una unidad a su cantidad total de municiones actuales.
- *Y button*. Ejecuta una animación de victoria por parte del personaje del jugador.
- *A button*. Este botón permite distintas acciones dependiendo el contexto. En el menú principal, la selección de una opción se hace a través de éste.

Dentro del juego mismo permite al usuario realizar dos tareas: tomar un ítem cercano y reanimar al compañero, en caso de que este se encuentre inactivo y a una distancia mínima.

- *B button*. Permite enviar una señal de ayuda al compañero del usuario. Esta acción puede incluso ser activada aún cuando el personaje del jugador se encuentra en un estado de inactividad. Además en otros contextos fuera del juego, sirve como una de cancelación o retroceso por las pantallas.

7.2.3. Control de pruebas

Adicionalmente, la aplicación permite algunas acciones a través del *gamepad* que son ajenas al control del juego. Estas tienen como objetivo facilitar el análisis de distintos módulos y situaciones útiles para verificar el modelo emocional propuesto.

- *Left bumper*. Realiza el cambio de cámara entre aquella que permite controlar al personaje y una cámara libre para observar desde cualquier ángulo el mundo virtual representado.
- *Left stick*. Permite el movimiento de la cámara libre sobre el plano definido por su vector de dirección y su posición.
- *Right stick*. Cambia la orientación de la cámara libre.
- *Left trigger y Righth trigger*. Permite el movimiento de la cámara libre sobre el eje Y.
- *Directional pad*. Dependiendo la dirección en que se presione, activa o desactiva la presencia de ciertos objetos como: el grafo de navegación, la ruta planeada, la información de metas activas o la interfaz de usuario.

7.3. Funcionamiento del sistema

7.3.1. Aplicación

Nuestro sistema cumple con gran parte de los requisitos y propiedades para considerarse un videojuego. Su diseño fue elaborado utilizando técnicas probadas en el área lo que nos permite asegurar su rendimiento. Algunas de nuestras pruebas indicaron que el número de cuadros por segundo que se dibujan en pantalla es reducido únicamente por una de las bibliotecas externas

que controla la animación de los personajes, factor que por supuesto puede ser mejorado en distintas maneras.

Otro de los factores que pueden afectar el rendimiento, es la generación de basura en memoria con el correspondiente uso del *garbage collector*. Este factor puede apreciarse ocasionalmente después de transcurrido un tiempo produciendo una leve interrupción en la imagen, fenómeno que puede ser causa de molestia para el jugador, sin embargo, su efecto no es tan frecuente debido a algunas de las técnicas que se utilizaron aunque se espera que en un trabajo futuro este problema pueda disminuirse aún más.

Experimentando con configuraciones del motor de inteligencia artificial y algunas otras características de las entidades en el entorno, observamos una gran variedad de aspectos que afectan la dificultad que puede crearse dentro del juego. Esto puede ser nocivo en la experiencia del usuario. Para solucionar esto, es posible ajustar algunos de los parámetros que alteran las características de los agentes y las de su entorno:

- *Carga de nivel*. Permite especificar el archivo que contiene los datos que conforman todas las entidades estáticas que se encuentran distribuidas en el entorno.
- *Máxima energía*. Determina el máximo nivel de energía que pueden alcanzar los agentes. Aplicada al avatar del usuario prevendría su eliminación temprana.
- *Máxima velocidad*. Establece la máxima velocidad a la cual pueden desplazarse los agentes. Aplicada a los enemigos incrementaría la dificultad al hacerlos más difíciles de evadir y disparar.
- *Daño*. Indica el daño causado por el arma principal de los agentes. Incrementando este parámetro al usuario facilitaría su tarea de eliminar a sus enemigos.
- *Precisión*. Para cada personaje a excepción del controlado por el jugador, la acción de disparo es producida con un nivel de precisión hacia la dirección del objetivo. Disminuyendo la precisión de los agentes ayudará al jugador a no ser golpeado con demasiada frecuencia.
- *Frecuencia de disparo*. Indica la cantidad de disparos que pueden realizarse por segundo. El incremento de este parámetro puede generar un grupo de proyectiles activos, de los cuales difícilmente un agente podrá escapar.
- *Lapso de memoria*. El tiempo máximo en el que cada personaje recordará a agentes enemigos previamente detectados por el sistema de visión.

- *Tiempo de resurrección.* Lapso de tiempo máximo en el cual los agentes enemigos reiniciarán su actividad después de ser eliminados.
- *Tiempo de reaparición de ítems.* Lapso de tiempo máximo para que cada ítem que fue tomado reaparezca nuevamente para su consumo.
- *Duración de una cubierta.* Tiempo máximo permitido a un personaje para utilizar la cubierta.

Pruebas realizadas con ajustes a estas variables permiten la creación de diversos niveles de dificultad, lo que a su vez, genera diferentes variaciones del entorno y los acontecimientos. Esto produce que los agentes puedan encontrarse en situaciones que, en un sistema más estático, no se presentarían con regularidad.

7.3.2. Identificación del modelo emocional

Atendiendo a la finalidad por la cual fue creada esta aplicación, verificamos el funcionamiento del modelo emocional y cómo es que éste afecta el comportamiento de los agentes en las distintas situaciones que se presentan.

Al igual que el ajuste de parámetros en la dificultad del juego, el modelo emocional requiere también de un conjunto de datos que deben proporcionarse para aquellas entidades emotivas presentes. En nuestro caso, estas variables afectarán únicamente al compañero del jugador. Su configuración, podría ser una forma para especificar ciertos tipos de personalidad.

Para cada emoción que pueda experimentar el personaje se debe especificar:

- *Umbral.* Valores bajos de umbral indicarán una facilidad del personaje por experimentar la emoción relacionada. Por otro lado, en una emoción como el estar contento, si su umbral está muy alto el personaje será poco susceptible a esta emoción, lo que podría interpretarse como un carácter pesimista, donde pocas veces se alegra de una situación.
- *Factor de decaimiento.* Indicando la rapidez con la que la emoción relacionada desaparecerá con el paso del tiempo. Un factor bajo de decaimiento en una emoción como el desprecio, puede indicar que un personaje mantiene su estado emotivo por largo tiempo, lo que podría representar una personalidad rencorosa que no olvida fácilmente los sucesos negativos.

Algunos otros parámetros pueden ser considerados por parte del generador de acontecimientos, para establecer las condiciones específicas sobre cada uno de los acontecimientos detectados. Sin embargo, en el caso de nuestra implementación nos sugiere una restricción sobre la cual debemos ser precavidos a un cambio de este tipo, ya que podría afectar no solo al rendimiento del sistema sino también una dificultad adicional al tratar de introducir otras entidades emotivas dentro del entorno.

Estableciendo los valores relacionados con la arquitectura afectiva realizamos una serie de observaciones con la finalidad de validar el funcionamiento del sistema afectivo. A continuación, se presentan los comportamientos que se presentaron en el caso de cada una de las emociones activas.

7.3.2.1. Un estado sin emoción

Una entidad emotiva puede encontrarse en un *estado emocional neutro*, es decir, un estado en el cual no tiene ninguna emoción. Lo que significa que la intensidad no ha superado el umbral en ninguna de las emociones posibles. En este caso, se considera un estado cognitivo donde la toma de decisiones generalmente se realiza por una prioridad dada a cada una de las metas que fue especificada previamente, nos referimos a los valores por defecto.

En nuestro diseño se asignó un orden a las metas de mayor prioridad. En un estado emocional neutro el agente atacará al enemigo mientras su energía este en un nivel aceptable y su compañero no necesite ayuda. Esta situación se presenta justo al iniciar el juego.

La meta del agente cambia en caso de que su compañero requiera de ayuda. Esta situación se aprecia en la Figura 7.9 donde su meta ha cambiado para reanimar a su compañero herido.



Fig. 7.9 Ninguna emoción activa. Meta de protección del equipo.

En otra situación, dado que su supervivencia es la meta de mayor prioridad, la Figura 7.10 muestra como en un estado de baja energía, el plan generado lleva al agente en busca de un ítem, ignorando por completo que su compañero esta herido y solicitando su ayuda.



Fig. 7.10 Ninguna emoción activa. Meta de supervivencia.

7.3.2.2. La emoción de contenido

Distintos acontecimientos han sucedido en el entorno y ahora nuestro personaje experimenta una emoción de júbilo definida en nuestro diseño como la emoción de *Contento*.

Su expresión es clara y sus intenciones también. En la Figura 7.11 se muestra como a pesar de que su estado no es el mejor pues las municiones de su arma están agotadas. Su plan es proteger a su equipo y debido a que su compañero no esta solicitando su ayuda, puede recoger municiones que mejoren la condición general del equipo.



Fig. 7.11 Emoción de Contento.
El personaje es feliz de proteger a su equipo

Su estado emotivo ha provocado que descuide su propia supervivencia dando preferencia a atacar al enemigo, lo que puede observarse en la Figura 7.12 pues después de recuperar sus municiones cambia sus planes por atacar, ignorando que su condición no es la mejor.



Fig. 7.12 La emoción de Contento lo hace arriesgar su energía

7.3.2.3. La emoción de desprecio

Por ultimo, analizamos una de las situaciones en las que la emoción presente es aquella a la que definimos como *Desprecio*. Experimentando una emoción de atribución esta indica que alguna de las normas planteadas no ha sido cumplida. En este caso, suponemos una situación donde la entidad emotiva fue alterada, quedando en estado de inactividad y a pesar de que la señal de ayuda fue enviada la ayuda no ha llegado rápidamente. La Figura 7.13 muestra como después del tiempo establecido el jugador decide ayudar a su compañero, el cual inmediatamente muestra su emoción actual e inicia un esperado plan para obtener un ítem de energía debido a su débil estado interno.

Mientras tanto el jugador es atacado por enemigos cercanos lo que debilita su estado. Analizando sus posibilidades el usuario solicita ayuda a su compañero el cual ahora se encuentra en mejor estado debido al ítem de energía que recién ha obtenido. Sin embargo, la emoción de desprecio hacia su compañero, aún se encuentra presente. Esto provoca un cambio en la jerarquía de las metas. Dando prioridad a su propia supervivencia. La Figura 7.14 demuestra como el plan trazado por el personaje afectivo corresponde a la obtención de otro ítem para mejorar sus posibilidades, ignorando por completo la señal de ayuda enviada previamente. Este comportamiento continúa, aún después de ser derrotado el jugador, situación que le deja como única oportunidad esperar la degradación de la emoción de *Desprecio* con el paso del tiempo. Finalmente cuando llegue este estado, si su compañero esta en condición de ayudarlo, lo hará.



Fig. 7.13 Emoción de desprecio al no ser ayudado con anterioridad.



Fig. 7.14 La emoción de desprecio continúa. Nótese que su energía es de 80% y aún así su meta es sobrevivir antes que otra cosa.



Conclusiones y Trabajo Futuro

El primer objetivo planteado en este trabajo fue relacionado al enriquecimiento de las conductas de agentes autónomos en ambientes virtuales. Esta idea planteó el uso de la teoría OCC, la cual pretende explicar que las emociones tienen un origen en los procesos cognitivos de los individuos. Las teorías e investigaciones analizadas en el capítulo 2 de este trabajo, mostraron la influencia de los elementos afectivos en el comportamiento de los humanos, específicamente en el proceso de toma de decisiones. Estas ideas fueron la base para plantear la hipótesis que el comportamiento de agentes autónomos sería enriquecido con una estructura cognitiva de las emociones para ser implementada, dentro del contexto de una arquitectura de videojuego. Siendo esta, una de las aportaciones más importantes.

Tomando como marco de nuestro trabajo, fueron considerados el tipo de agentes que *habita* los videojuegos, estos se ven relacionados con un conjunto de acontecimientos y objetivos propios establecidos por el videojuego. Dado que el actor principal es el usuario, un individuo naturalmente emocional, este trabajo pretendió mejorar su relación con los agentes autónomos presentes, a través de la creación de comportamientos emergentes, cuyo origen sean las emociones experimentadas por el agente. Estas emociones se encontrarán predefinidas por el diseño del videojuego y activadas por la interacción con el entorno, el usuario y los objetivos de cada uno.

Para poder incluir las emociones dentro del comportamiento de los agentes, planteamos el objetivo de crear una arquitectura afectiva, la cual debía satisfacer ciertos aspectos básicos relacionados con las características de los sistemas que forman parte de un videojuego.

El diseño del juego fue afectado con la única finalidad de incorporar los elementos clave, tanto en el comportamiento final de los agentes como en el reconocimiento de ciertas condiciones necesarias en el entorno. Esta etapa

ofreció un análisis de los personajes, enriquecido por la exploración tanto de sus metas como de los comportamientos, siendo todos estos afectados por distintas emociones. Éstas fueron definidas de manera formal para evitar muchos de los problemas de ambigüedad que frecuentemente se encuentran en la interpretación de las emociones.

Creemos que la formalización de las emociones aún requiere de un refinamiento que permita definir de manera clara su dinámica y funcionamiento con otros aspectos del agente. La reorganización que se sugiere sobre la jerarquía de las metas se plantea como un hecho inamovible en cada una de las emociones. Aunque es inevitable indicar ciertos aspectos de esta organización, se espera la existencia de una mejor técnica para realizar este proceso de forma más flexible. Permitiendo, con base en un conjunto de parámetros, justificar este cambio realizado en tiempo de ejecución. Esta idea podría traer a los agentes un sistema de metas dinámico que evoluciona con el tiempo adoptando nuevas metas y desechando otras.

Otro aspecto a considerar es el referente a la personalidad de cada agente. Aunque este es un fenómeno que se ha conseguido simular de muchas maneras diferentes, por ejemplo con la variación de parámetros; una extensión de nuestra arquitectura podría plantear un módulo de personalidad que influya en aspectos como la tendencia a experimentar cierto grupo de emociones, sin embargo, su aportación más destacada será agregando una etapa extra a la toma de decisiones. En el caso de que un agente presente una emoción en un determinado momento, como puede ser la tristeza, un módulo de personalidad puede ayudar a formalizar un plan de acción. Esto es, si se considera que un agente tiene una personalidad depresiva, la tristeza experimentada puede ocasionar que pierda la mayoría de sus metas activas. Produciendo un comportamiento de inactividad. De manera contraria si el agente tiene una personalidad alegre, una emoción de tristeza puede transformarse en una motivación para tomar un plan de acción que elimine la causa de tal estado afectivo, ignorando inclusive metas ya existentes.

La memoria es otra de las variables que deben considerarse para una extensión del modelo. Su uso en forma de recuerdos puede ayudar a crear tendencias afectivas en los personajes por acontecimientos sucedidos anteriormente. En una etapa inicial este efecto puede alterar de manera progresiva los umbrales especificados por cada emoción; lo que también ofrecerá una nueva manera de crear una personalidad cambiante en los personajes. Gradualmente esta memoria afectiva también podría crear cambios en funciones que determinan el grado de impacto en la valoración de cada acontecimiento. Haciendo al agente susceptible a ciertos sucesos del entorno que antes eran ignorados.

Existen distintos fenómenos afectivos que nuestra arquitectura no toma en cuenta, y aunque consideramos que una integración progresiva de cada uno de estos aspectos es posible, el estado actual de nuestra implementación ofrece un

cambio sutil en el comportamiento, apreciable en distintas situaciones que se presentaran de diferente manera al cambiar el juego o el rango y tipo de las emociones que pueden experimentar.

Consideramos que una de las aportaciones que ofrece éste trabajo esta relacionado con la formalización de la arquitectura afectiva, la cual ha diferencia de otras investigaciones, y en el caso de estudio trata de apegarse más a los requerimientos y necesidades de los videojuegos. Esto se logra considerando las restrictivas condiciones que comúnmente tiene el módulo de inteligencia artificial dentro del sistema de un videojuego. Además, la arquitectura ofrece una herramienta de diseño que permite a los desarrolladores guiarse en la construcción de los comportamientos influenciados por las emociones de cada agente. Aunque creemos que la arquitectura afectiva cumplió con nuestras necesidades sabemos que aún puede refinarse e incluir elementos que no se consideraron dentro de la realización de este trabajo.

La importancia de implementar nuestra arquitectura afectiva, se centra en el diseño de un sistema que cumple con la mayoría de las características que se identifican en un videojuego. Este tenía como objetivo ofrecer un entorno donde el usuario pudiera interactuar con otros agentes cumpliendo objetivos específicos. Uno de estos agentes fue diseñado para experimentar dos tipos de emociones, las cuales afectan su comportamiento de distinta manera. Gracias a la ayuda de distintas herramientas, este trabajo concluyo con un sistema funcional que satisfactoriamente incorporo los distintos componentes de la arquitectura afectiva, además del uso de técnicas comunes en el desarrollo de videojuegos.

En un trabajo futuro es posible incorporar aspectos que en conjunto con el uso de las emociones y sus consecuencias en el proceso de toma de decisiones ayudará a mejorar la ilusión de realismo de los agentes. Estos no fueron cubiertos en la implementación dados los escasos recursos que se tenían a disposición y la complejidad tanto técnica como artística que se requiere. Algunos de estos son:

- *Modelos y Animación.* Modelos más detallados, así como un rango más amplio de animaciones y una fluidez en ellas ayudará sustancialmente a incrementar la credibilidad de cada uno de los personajes. Técnicas como *skinning* y *normal mapping* son usadas actualmente en videojuegos comerciales justamente para crear estos efectos.
- *Expresión facial.* Aunque la interfaz creada en este proyecto muestra una forma muy primitiva de esta característica, la expresión facial aplicada directamente a los modelos presentes en el entorno es imprescindible para generar una sensación de comunicación directa con los agentes.

- *Audio.* Aunque existen muchas y distintas formas para que los agentes se comuniquen, una de las formas más naturales y humanas es a través del lenguaje hablado. Técnicas que incluyan el uso de diálogos por parte de los agentes será un elemento muy importante para los usuarios, más aún si las voces de los agentes son reguladas por las emociones que experimenten.

También pueden incluirse mejoras al sistema relacionadas con la arquitectura afectiva, las cuales podrían generar resultados distintos a los expuestos.

- *Percepción y Ejecución.* La inclusión de las emociones en aspectos como la percepción del entorno o de sí mismo, y la ejecución de acciones permitirá a los agentes expresar con más claridad los efectos del estado afectivo en el que se encuentren.
- *Efecto Répate.* Prestando más atención a este efecto, pueden hacerse mediciones sobre las emociones, ajustando el nivel de dificultad. De esta manera el usuario toma en cuenta de manera efectiva los estados afectivos pues éstos pueden tener mayor relevancia en el desarrollo del juego.

Encontramos investigaciones que demuestran la importancia de las emociones en la interacción social. Creemos que nuestra implementación puede ayudarnos a observar este fenómeno. Al experimentar con el videojuego creado, las distintas situaciones establecen una relación entre el jugador y su compañero fomentada por la capacidad de éstos de solicitar su ayuda mutua. El uso de estados afectivos en esta relación ocasiona que el usuario experimente una inmersión mayor al tener una dependencia sobre su compañero y preocuparse por sus emociones, ya que de ellas depende el comportamiento. Y este último puede ser de ayuda en la consecución de sus metas o en otro sentido producir un escenario con contenga una mayor dificultad para la consecución de las metas.

Esperamos que en un trabajo futuro las ideas de nuestra arquitectura puedan ser probadas en un videojuego comercial. Existen videojuegos que ofrecen lenguajes script o módulos externos que permiten controlar los personajes dentro del juego, sin embargo, creemos que uno de los retos sobre estas plataformas se encuentra en la detección y procesamiento de acontecimientos. Otra idea sugeriría la creación de un nuevo juego con una temática y estilo diferente. Un juego deportivo puede permitir la existencia de personajes con vidas más largas con lo que eventualmente pueden cambiar sus emociones y metas. Enfatizando el camino, por ejemplo, un basquetbolista, para convertirse en profesional dependería de otros jugadores a los que enfrente y de sus propios compañeros. El juego puede incluso sugerir que si el jugador saca de su equipo a un compañero este tal vez se muestre rencoroso en un futuro.

Este tipo de juego agregaría un nivel más de dificultad al preocuparse de las emociones de otras personas aún persiguiendo metas propias, como en la vida real. Esto puede sugerir una plataforma de prueba en la que se puede buscar que tipo de juegos fomentan el trabajo en equipo.

Más aún, en el camino de dar una verdadera importancia a las emociones tanto de los usuarios como de los agentes, dispositivos como *Emotiv neuroheadset* [EMOS08] pueden ayudar a extender el sistema permitiendo que los agentes puedan recibir información sobre el estado afectivo del jugador creando metas y acontecimientos que produzcan una relación profunda y más humana.

Finalmente, aunque sabemos que aún hay mucho trabajo por realizar, creemos que la meta de este proyecto fue cumplida, dejándonos como producto un sistema, similar a un videojuego, que cumplió con todos nuestros requerimientos. Además el análisis y diseño del comportamiento resultó de utilidad para observar de manera más clara la relación de las emociones y su influencia en el proceso de toma de decisiones.



Apéndice A

Concepto de juego: *Dave Wars*.

Introducción

Dave Wars es un juego de disparos en tercera persona para *Xbox 360* que lleva al jugador a pelear junto a un compañero En una realidad alterna donde personajes famosos se encuentran reunidos por un mismo fin, demostrar quien es el mejor.

Antecedentes

El modo del juego esta inspirado en juegos como *Gears of War* y *Halo 3* donde el jugador controla al personaje principal y es seguido por miembros de su equipo. El jugador tiene muy poco control sobre su equipo lo que le impide de forma fácil eliminar grandes cantidades de enemigos.

Por otro lado, la idea alrededor de este juego consiste en mezclar distintos personajes conocidos y verlos en situaciones nuevas.

Género

Juego de disparos en tercera persona y acción táctica alrededor de una historia de ciencia ficción.

Descripción

Dentro del juego el usuario controla al personaje principal, el cual es acompañado por otro que representa su compañero. Ambos forman un equipo que tiene como objetivo eliminar a todo enemigo que se cruce en su camino.

El jugador y su compañero podrán correr, recolectar objetos, usar cubiertas disponibles en el ambiente, levantar al compañero herido y disparar su arma principal para eliminar a sus enemigos. Además podrán mandar señales de ayuda en caso de que se encuentren en peligro o hayan sido derrotados, sin embargo cada miembro del equipo tendrá la facultad de decidir si atender este llamado o continuar con sus objetivos. El jugador deberá preocuparse no solo por la integridad de su equipo sino también por los sentimientos de su compañero, evitando su molestia. Siendo constante la meta de sobrevivir al ataque enemigo.

Cada personaje cuenta con un arma disponible que requiere de la recolección de municiones para que pueda funcionar. Además deben preocuparse de su energía la cual decae con cada ataque enemigo y puede ser recuperada recolectando el objeto adecuado. Cuando la energía de los enemigos es agotada estos caerán inactivos. Transcurrido un tiempo se levantarán para reanudar su objetivo (eliminar elementos del equipo contrario).

El juego terminara cuando el jugador y su compañero agoten su energía. Y esto último trae como consecuencia la imposibilidad de ayudarse mutuamente.

Características principales

El principal elemento de este juego se encuentra en el manejo de la inteligencia artificial de los personajes, donde se pretende incorporar varios modelos tradicionales, como un sistema de metas jerárquico, y un nuevo modelo de emociones con el objetivo de dar mayor realismo a las conductas y toma de decisiones de los *NPC (Non Player Characters)*

Plataforma

El juego esta enfocado principalmente a jugarse en el *Xbox 360* vía el *XNA Framework* pero se planea que tenga la versatilidad para usarse en una PC sin mucha capacidad gráfica.



Referencias

[ACED08] Acevedo-Moreno David Aurelio, Laureano-Cruces Ana Lilia, Ramírez-Rodríguez Javier. *Design of an Emotive Architecture for Videogames*, Memorias en CD, ISBN 978-970-15-14388-2, pp 99-106, En el XXI Congreso Nacional y VII Congreso Internacional de Informática y Computación de la ANIEI. Monterrey 1-3 de octubre del 2008.

[AHNH06] Ahn, H.I., Picard R.W. *Affective Cognitive Learning and Decision Making: The Role of Emotions*, The 18th European Meeting on Cybernetics and Systems Research, Vienna, Austria, 2006

[ALLJ02] Allbeck Jan, Badler Norman, *Toward Representing Agent Behaviors Modified by Personality and Emotion*, Embodied Conversational Agents at AAMAS02, 2002

[ARRI01] Arriaga, F., El Alami, M., Ugena A., *Acceleration of the Transfer of Novices into Experts: the Problem of Decision Making*, Proceedings International Conference BITE'01, University of Eindhoven 2001, pp.245-258.

[ATIA08] Productos y Tecnologías de ATI, Advanced Micro Devices, Inc. 2008
<http://ati.amd.com/sp/products/index.html>
Consultado en Noviembre 2008

[AUTK08] Autodesk Kynapse
<http://usa.autodesk.com/adsk/servlet/index?id=11390544&siteID=123112>
Consultado en Noviembre 2008

[BACJ06] Bach Joscha, Dörner Dietrich, Vuine Ronnie, *Psi and MicroPsi: A Novel Approach to Modeling Emotion and Cognition in a Cognitive Architecture*, Tutorial at ICCM 2006

[BECC05] Becker, C., Nakasone, A., Prendinger, H., Ishizuka, M., Wachsmuth, I. *Physiologically interactive gaming with the 3D Agent Max*. International Workshop on Conversational Informatics, 2005

[BECC06] Becker Christian, Lessman Nadine, Kopp Stefan, Wachsmuth Ipke, *Connecting Feelings and Thoughts – Modeling the Interaction of Emotion and Cognition in Embodied Agents*, Faculty of Technology, University of Bielefeld, Alemania, 2006

[BERJ97] Berry J., Poortinga Y., Pandey J., Dasen P., Saraswathi T., *Handbook of Cross-cultural Psychology: Basic Processes and Human Development*, Editorial Allyn & Bacon, Segunda Edición, 1997, ISBN: 978-020-51-6075-4, pág. 145.

[BETT08] Bettcher Matt, *Using a Pool memory structure to boost performance*, Ziggyware XNA news and Tutorials, 2008
http://www.ziggyware.com/readarticle.php?article_id=213
Consultado en Noviembre 2008

[BUCM05] Buckland Mat, *Programming Game AI by Example*, Wordware Publishing, 2005, ISBN: 978-155-62-2078-4

[BUNG08] Bungie, 2008
<http://www.bungie.net/>
Consultado en Enero 2008

[BURR01] Burke, R., Isla, D., Downie, M., Ivanov, Y., Blumberg, B. *CreaturesSmarts: The Art and Architecture of a Virtual Brain*, Proceedings of the Game Developers Conference, International Game Developers Association, 2001, pp. 147-166.

[BURS08] Bura Stéphane, *Emotion Engineering: A Scientific Approach For Understanding Game Appeal*, Gamasutra, Julio 29, 2008
http://www.gamasutra.com/view/feature/3738/emotion_engineering_a_scientific_.php
Consultado en Agosto 2008

[CASJ00] Cassell Justin, Sullivan Joseph, Prevost Scott, Churchill Elizabeth, *Embodied Conversational Agents*, The MIT Press, 2000, ISBN: 978-026-20-3278-0

[CASS99] Castañeda, S. Martínez, R. *Enseñanza y aprendizaje estratégicos: Modelo integral de evaluación e instrucción*. En Revista Latina de Pensamiento y Lenguaje. 4:2B, 1999, pp. 251-278.

- [CHAA07] Champandard Alex, *Top 10 Most Influential AI Games*, AiGameDev.com, Septiembre 12, 2007
<http://aigamedev.com/reviews/top-ai-games>
Consultado en Febrero 2008
- [CHAA07₂] Champandard Alex, *Living with The Sims' AI: 21 Tricks to Adopts for Your Game*, AiGameDev.com, October 15th, 2007
<http://aigamedev.com/reviews/the-sims-ai>
Consultado en Febrero 2008
- [CHOE02] Chown Eric, Jones Randolph M., Henninger Amy E., *An Architecture for Emotional Decision-Making Agents*, En Proceeding of the first international joint conference on Autonomous agents and multiagent systems:part 1, Bologna, Italia, 2002, ISBN: 1-58113-480-0
- [CHRM02] Christian Mike, *A Simple Inference Engine for a Rule-Based Architecture*, AI Game Programming Wisdom, Charles River Media, 2002, ISBN: 978-158-45-0077-3
- [COOD07] Cook Daniel, *Constructing Artificial Emotions: A Design Experiment*, Gamasutra, Octubre 26, 2007
http://www.gamasutra.com/view/feature/1992/constructing_artificial_emotions_.php
Consultado en Enero 2008
- [CHAA03] Champandard Alex. *AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors*, New Riders Publishing, 2003, ISBN: 978-159-27-3004-9
- [DAMA94] Damasio, A. R. *Descarte's error: Emotion, reason, and the human brain*. Editorial Avon Books, 1994, pp. 70, ISBN: 978-038-07-2647-9
- [DELM00] Deloura Mark, *Game Programming Gems*, Charles River Media, 2000, ISBN: 978-158-45-0049-0
- [DIRX08] DirectX Developer Center, Microsoft Corporation, 2008
<http://msdn.microsoft.com/en-us/directx/default.aspx>
Consultado en Febrero 2008
- [EASP08] EA Sports, 2008
<http://www.easports.com/>
Consultado en Febrero 2008
- [EBED06] Eberly David H., *3D Game Engine Design, Second Edition: A Practical Approach to Real-Time Computer Graphics*, Morgan Kaufmann, Segunda Edición, 2006, ISBN: 978-012-22-9063-3

- [EMOS08] Emotiv Systems, EmotivEPOC, 2008
<http://emotiv.com/>
Consultado en Marzo 2009
- [ESA07] Entertainment Software Association, *Essential Facts About The Computer And Video Game Industry*, 2007
<http://www.theesa.com/>
Consultado en Febrero 2008
- [ESTE02] Estévez-Nénninger E.H. *Enseñar a Aprender: estrategias cognitivas*. Colección Maestros y Enseñanza. Editorial Piados. México-Buenos Aires-Barcelona. 2002. ISBN: 9-68853-502-8
- [FACA08] Façade, a one-act interactive drama, 2008
<http://interactivestory.net/>
Consultado en Enero 2008
- [FRED03] Freeman David, *Creating Emotion in Games: The Craft and Art of Emotioneering*, New Riders Publishing, 2003, ISBN: 978-159-27-3007-0
- [GANS93] Gandevia Simon, Burke David, Anthony Michael, *Science and Practice in Clinical Neurology*, Editorial Cambridge University Press, 1993, pág. 3, ISBN: 978-052-14-3119-4
- [GARA08] García Rojas Alejandra, Gutiérrez Mario, Thalmann Daniel, *Simulation of Individual Spontaneous Reactive Behavior*, En Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems, Volume 1, Estoril, Portugal 2008, pp. 143-150, ISBN: 978-098-17-3810-9
- [GILK04] Gilleade Kiel M., Dix Alan, *Using Frustration in the Design of Adaptive Videogames*. En Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology, Singapore, 2004, pp. 228-232, ISBN: 1-58113-882-2
- [GILK05] Gilleade K. M., Dix A., Allanson J., *Affective videogames and modes of affective gaming: Assist me, challenge me, emote me*. In The 2005 International Conference on Changing Views: Worlds in Play, 2005
- [GIND00] Ginsburg Dan, *Octree Construction*, Game Programming Gems, Charles River Media, 2000, pp. 439-443, ISBN: 978-158-45-0049-0
- [GOLD97] Goleman Daniel, *Emotional Intelligence: Why it Can Matter More Than IQ*, Editorial Bantam, 1997, ISBN: 978-055-33-7506-0
- [GOMJ05] Gómez Álvarez José Manuel, *Sistemas de Información Multiagente*, Programa de Doctorado: Retos Científicos de la Computación, Septiembre 2005

http://www.irisel.com/~jmgomez/IT/doctorate/taller_2.htm

[GONM98] Gonzalez M.P., Barrul E., Pons C., Marteles P., *¿Qué es la emoción?*, 1998
http://www.biopsychology.org/biopsicologia/articulos/que_es_la_emocion.htm
 Consultado en Febrero 2009

[GRAJ04] Gratch J., Marsella S., *A domain-independent framework for modeling emotion*, Cognitive Systems Research 5, 2004, pp. 269-306.

[HALO07] Halo 3, 2007
<http://www.halo3.com/>
 Consultado en Enero 2008

[HARM04] Harmon Matthew, *A System for Managing Game Entities*, Game Programming Gems 4, Charles River Media, 2004, pp. 69-83. ISBN: 978-158-45-0295-1

[HERC07] Hermann C., Melcher H., Rank S., Trappl R. *Neuroticism - A Competitive Advantage (Also) for IVAs?* En Pelachaud C. et al. (eds.), Intelligent Virtual Agents, Springer, Berlin/Heidelberg/New York, 2007, pp.64-71, ISBN: 978-354-07-4996-7

[HERE08] Hernández Emmanuel, *Arquitectura de Inteligencia Artificial para Sistemas Multiagente Colaborativos*, Tesis de Maestría, UNAM, 2008

[HIGD02] Higgins Dan, *Pathfinding Design Architecture*, AI Game Programming Wisdom, Editor Steve Rabin, Charles River Media, 2002, pp. 122-132, ISBN: 978-158-45-0077-3

[INTE08] Inteligencia Emocional, 2008
<http://www.inteligencia-emocional.org>
 Consultado en Enero 2008

[JAQP04] Jaques, P. A. *Using an Animated Pedagogical Agent to Interact Affectively with the Student*. Universidade Federal Do Rio Grande Do Sul, 2004

[JAQP042] Jaques , P. and Viccari, R., 2004. *A BDI Approach to Infer Student's Emotions*. IBERAMIA 2004, Lecture Notes on Artificial Intelligence 3315, Springer Verlag Berlín Heidelberg, pp. 901-911, ISBN: 978-354-02-3806-5

[KENF02] Kenneth D. Forbus, *Under the Hood of the Sims*, CS 395 Game Design, 2002,
http://www.cs.northwestern.edu/~forbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/v3_document.htm
 Consultado en Febrero 2008

- [KURA07] Kuri Morales Ángel, Galaviz Casas José, *Algoritmos genéticos*, Segunda Edición, Sociedad Mexicana de Inteligencia Artificial, 2007, ISBN:968-16-6383-7
- [LAUA00] Laureano-Cruces, A. *Interacción dinámica en sistemas de enseñanza inteligentes*, Tesis de Doctorado en Investigación Biomédica Básica, Instituto de Investigaciones Biomédicas – UNAM. (2000) <http://delfosis.uam.mx/~ana/>. 2000.
- [LAUA00₂] Laureano-Cruces, A., de Arriaga-Gómez, F., *Reactive Agent Design For Intelligent Tutoring Systems*, *Cybernetics and Systems*, Vol. 31, No. 1, pp. 1-47, 2000.
- [LAUA01] Laureano-Cruces, A. L., de Arriaga, F., García-Alegre Sánchez, M.C. *Cognitive Task Analysis: A proposal to Model Reactive Behaviors*, *Journal of Experimental & Theoretical Artificial Intelligence*, 2001, 13:227-239
- [LAUA03] Laureano-Cruces, A. L., Barceló-Aspeitia A., *Formal Verification of Multi-Agent System Behaviour Emerging from Cognitive Task Analysis*. En *Journal of Experimental & Theoretical Artificial Intelligence*. Ed. TAYLOR & FRANCIS, Reino Unido, 2003, 15:407-413.
- [LAUA04] Laureano-Cruces Ana Lilia, *Agentes Pedagógicos*, Memorias en CD, ISBN: 970-36-0155-3. En el XVII Congreso Nacional y III Congreso Internacional de Informática y Computación de la ANIEI. Tepic, Nayarit, 20-22 de octubre del 2004.
- [LAUA05] Laureano-Cruces Ana Lilia, Espinosa-Paredes Gilberto, *Behavioral Design to Model a Reactive of an Expert in Geothermal Wells*. En *International Journal of Approximate Reasoning*, Elsevier, USA., Vol. 39, No.1 pp, 1-28, abril 2005.
- [LAUA06] Laureano-Cruces Ana Lilia, *Emociones Sintéticas y Avatars*, De la Reflexión a la Acción, Editado por la División de Ciencias y Artes para el Diseño. CYAD / UAM-A. 2006, pp. 251-255.
- [LAUA08] Laureano Cruces A., Ramirez-Rodriguez J., Mora-Torres M., Arriaga F., Escarela-Pérez R., *Cognitive-Operative Model of Intelligent Learning Systems Behavior*, *Interactive Learning Environments*, Elsevier, USA, 2008, ISSN: 0888-613X. Publicado en línea:
<http://www.informaworld.com/smpp/content~content=a795273447~db=all~order=pubdate>
- [LAUA08₂] Laureano-Cruces Ana Lilia, Sánchez-Guerrero Lourdes, Mora-Torres Martha, Ramírez-Rodríguez Javier. *Learning Objects and Personalized Instruction*, *Proceedings de E-Learn 2008 World Conference on E-Learning in*

Corporate Government, Healthcare, & Higher Education, pp. 1728-1736, November 17-21, Las Vegas, Nevada, USA. 2008.

[LAUA09] Laureano-Cruces Ana Lilia, Acevedo-Moreno David Aurelio, *Las emociones en la toma de decisiones: análisis y diseño de un caso concreto considerando la estructura cognitiva*, Aceptado. En el Libro: De la Evidencia Empírica a la Apropiación Tecnológica en Psicología. UNAM. 2009.

[LAUC09₂] Laureano-Cruces Ana Lilia, Velasco-Santos Perla, Mora-Torres Martha, Acevedo-Moreno David, *Hacia Interfaces Inteligentes*, Aceptado. En el Libro: Avances de las Mujeres en las Ciencias, las humanidades y todas las disciplinas. Sustentabilidad, Aire, Agua, Fuego y Tierra. CYAD/UAM-A. 2009.

[LAZN04] Lazzaro Nicole, *Why We Play Games: Four Keys to More Emotions Without Story*, Player Experience Research and Design for Mass Market Interactive Entertainment, XEODesign, Inc, 2004

[LIOH08] LionHead Studios
<http://www.lionhead.com/>
 Consultado en Enero 2008

[LOEG03] Loewenstein, G., Lerner, J.S, *The role of affect in decision making*. Handbook of Affective Sciences, Oxford New York: Oxford University Press, 2003, pp. 619-642, ISBN: 978-019-51-2601-3

[LONS95] Loncaric Sven, *Virtual reality foundations*, Faculty of Electrical Engineering and Computing, University of Zagreb, 1995

[LOPR08] López Rubén, *XNA Pk3Loader*, XNA CoderNet, 2008
<http://www.codernet.es/contents.html>
 Consultado en Enero 2008

[LUA08] Lua, The programming language.
<http://www.lua.org/>
 Consultado en Noviembre 2008

[MALM04] Malfaz María, Salichs Miguel A. *Design of an architecture based on emotions for an autonomous robot*, Carlos III University of Madrid, España, 2004

[MARJ08] Marcellin Jacques Sergio, Notas del Curso: Construcción de Sistemas Expertos, Posgrado en Ciencia e Ingeniería de la Computación, UNAM, 2008

[MARS01] Marsella S., Gratch J., *Tears and Fears: Modeling emotions and emotional behaviors in synthetic agents*, Proceedings of the Fifth International Conference on Autonomous Agents, 2001

[MARS02] Marsella S., Gratch J., *A step toward irrationality: using emotion to change belief*, En International Conference on Autonomous Agents: Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna, Italia, 2002. ACM Press. pp. 334-341, ISBN: 1-58113-480-0

[MARS04] Marsella S., Gratch J., *Evaluating a General Model of Emotional Appraisal and Coping*, AAAI Spring Symposium on Architectures for Modeling Emotion: Cross-disciplinary Foundations, Palo Alto, CA, 2004

[MARS06] Marsella S., Gratch J. *The Architectural Role of Emotion in Cognitive Systems*, Integrated Models of Cognitive Systems, Oxford University Press, 2006, ISBN: 978-019-80-4077-4

[MATJ02] Matthews James, *Basic A* Pathfinding Made Simple*, AI Game Programming Wisdom, Editor Steve Rabin, Charles River Media, 2002, pp.105-113. ISBN: 978-158-45-0077-3

[MCGC09] Microsoft Corporation, *Garbage Collection*, .Net Framework Developer's Guide, 2009
<http://msdn.microsoft.com/en-us/library/0xy59wtx.aspx>
Consultado en Enero 2009

[MCHL09] High Level Shading Language Documentation, Microsoft Corporation, 2009
[http://msdn.microsoft.com/en-us/library/bb509561\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb509561(VS.85).aspx)
Consultado en Enero 2009

[MRBS08] Mr. Bubb in Space, 2008
<http://www.zoesis.com/mrbubb/>
Consultado en Abril 2008

[MUEE85] Mueller Erik, Dyer Michael, *Daydreaming in humans and computers*, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, 1985

[NARA04] Nareyek Alexander, *AI in Computer Games*, Game Development, Vol. 1, No. 10, Febrero 2004

[NARA07] Nareyek Alexander, *Game AI Is Dead. Long Live Game AI!*, IEEE Intelligent Systems Vol. 22, No. 1, January/February 2007

[NVID08] Nvidia Corporation.
<http://www.nvidia.com/page/home.html>
Consultado en Noviembre 2008

[OBRJ02] O'Brien John, *A Flexible Goal-Based Planning Architecture*, AI Game Programming Wisdom, Editor Steve Rabin, Charles River Media, 2002, ISBN: 978-158-45-0077-3

[OPEG08] OpenGL, The Industry's Foundation for High Performance Graphics
<http://www.opengl.org/>
 Consultado en Noviembre 2008

[ORKJ05] Orkin Jeff, *Agent Architecture Considerations for Real Time Planning in Games*, En Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference, 2005

[ORKJ06] Orkin Jeff, *Three States and a Plan: The A.I. of F.E.A.R. Game*, En Proceedings of the Game Developer's, 2006

[ORTA88] Ortony A., Clore G, Collins A. *The cognitive structure of emotions*, Cambridge, UK: Cambridge University Press, 1988, ISBN: 978-052-13-8664-7

[PADC08] Padilla Castañeda Miguel A., Savage Jesús, Hernández Adalberto, Arámbula Cosío Fernando, *Local Autonomous Robot Navigation using Potential Fields*, Motion Planning, En Tech Education and Publishing, 2008, pp. 1-22, ISBN: 978-953-76-1901-5

[PAIA08] Paiva A., André E., Arafa Y., Costa M., Figueiredo P., Gebhard P., Höök P., Mamdani A., Martinho C., Mourão D., Petta P., Sengers P., Vala M., *SAFIRA – Supporting Affective Interactions in Real-time Applications*,
<http://www.sics.se/safira/>
 Consultado en Noviembre 2008

[PARR02] Parent Rick, *Computer Animation. Algorithms and Techniques*, Morgan Kaufman Publishers, 2002, ISBN: 978-155-86-0579-4

[PETP01] Petta Paolo, Trappl Robert, *Emotions and Agents*, Multi-agents systems and applications, Springer-Verlag, New York, USA, 2001, pp. 301-316, ISBN: 3-540-42312-5

[PETP97] Petta Paolo, Trappl Robert, *Why to Create Personalities for Synthetic Actors*, Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents, Spinger-Verlag, London, UK, 1997, pp. 1-8, ISBN: 3-540-62735-9

[PEIF05] Peinado Federico, *Diseño – Narración Interactiva*
 Presentaciones del curso de Diseño de Juegos, Departamento de Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid, 2005
<http://federicopeinado.com/>
 Consultado en Noviembre 2008

- [PERD08] Pearl Demon, 2008
<http://www.zoesis.com/pearldemon.html>
Consultado en Abril 2008
- [PICR97] Picard R. W., *Affective Computing*, MIT Press, Cambridge Boston London, 1997
- [PHYN08] PhysX by Nvidia.
http://www.nvidia.com/object/nvidia_physx.html
Consultado en Noviembre 2008
- [PREA08] Presagis AI.implant,
<http://www.presagis.com/products/simulation/details/aiimplant>
Consultado en Noviembre 2008
- [PYTH08] Python Programming Language.
<http://www.python.org/>
Consultado en Noviembre 2008
- [RABS00] Rabin Steve, *Designing a General Robust AI Engine*, Game Programming Gems, Charles River Media, 2000, pp. 221-236, ISBN: 978-158-45-0295-1
- [RABS02] Rabin Steve, *An Extensible Trigger System for AI Agents, Objects, and Quests*, Game Programming Gems 3, Charles River Media, 2002, pp. 285-293, ISBN: 978-158-45-0233-3
- [RAM07] Research And Markets, *Mexico Video Games Market Report 2007*
<http://www.researchandmarkets.com>
Consultado en Enero 2008
- [RANS04] Rank Stefan, Petta Paolo, Trappl Robert, *Features of Emotional Planning in Software Agents*, Austrian Research Institute for Artificial Intelligence, Vienna, Austria, 2004
- [RANS05] Rank Stefan, Lucas dos Anjos Pablo, Petta Paolo, Aylett Ruth, *What is In an Affective Architecture for Situated Agents*, Austrian Research Institute for Artificial Intelligence, Vienna, Austria, 2005
- [REEJ07] Reeve Johnmarshall, *Motivación y Emoción*, Editorial McGraw Hill, 3a. Edición, 2007, ISBN: 8-44811-825-1
- [ROLE99] Rolls E. T., *The brain and emotion*, New York: Oxford University Press, 1999, pp 282, ISBN: 978-019-85-2463-2

[RUSS04] Russel S. J., Norving P. *Inteligencia Artificial. Un enfoque moderno*, Segunda edición, Pearson Educación, Madrid, 2004, ISBN: 978-842-05-4003-0

[SAVJ04] Savage Jesús, Vázquez Gabriel, *Diseño de Microprocesadores*, Facultad de Ingeniería, Ciudad Universitaria, México, 2004

[SCHB04] Schwab Brian, *AI Game Engine Programming*, Charles River Media, 2004, ISBN: 978-158-45-0344-6

[SCHK00] Scherer, K. R. *Psychological models of emotion*. En *The neuropsychology of emotion*, Editado por Borod Joan, Oxford/New York: Oxford University Press. 2000, pp. 137-162, ISBN: 978-019-51-1464-5

[SCHK05] Scherer, K. R., *What are emotions? And how can they be measured?*, *Social Science Information*, Volume 44(4), 2005, pp. 695-729.

[SIMH67] Simon, H. A., *Motivational and emotional controls of cognition*, *Psychological Review*, 1967, pág. 74.

[SMIC90] Smith C., Lazarus R. *Emotion and adaptation*, *Handbook of personality: Theory & research*, Editado por Lawrence A. Pervin y Oliver P. John, New York: Guilford Press, 1990, pp. 609-637, ISBN: 978-157-23-0695-0

[SQUE08] Square-Enix
<http://www.square-enix.com/na/>
 Consultado en Noviembre 2008

[SCOB02] Scott Bob, *The Illusion of Intelligence*, *AI Game Programming Wisdom*, Editado por Rabin Steve, Charles River Media, 2002, pp. 16-20, ISBN: 978-158-45-0077-3

[STOB00] Stout Bryan, *The Basics of A* for Path Planning*, *Game Programming Gems*, Editado por DeLoura Mark, Charles River Media, 2000, pp.254-263, ISBN: 978-158-45-0049-0

[SWET98] Sweeny Tim, *UnrealScript Language Reference*, Epic MegaGames, Inc, 1998.
<http://udn.epicgames.com/Three/UnrealScriptReference.html>
 Consultado en Noviembre 2008

[SXMT09] Softimage|XSI Mod Tool, Autodesk Inc. 2008
<http://www.softimage.com/products/modtool/>
 Consultado en Enero 2008

[SYKJ04] Sykes Jonathan, *Affective gaming: advancing the argument for game-based learning*, eMotion Laboratory, Glasgow Caledonian University, UK

[THOA02] Thor Alexander, *An Optimized Fuzzy Logic Architecture for Decision-Making*, AI Game Programming Wisdom, Editado por Ravin Steve, Charles River Media, 2002, pp. 367-374. ISBN: 978-158-45-0077-3

[TOZP02] Tozour Paul, *Introduction to Bayesian Networks and Reasoning Under Uncertainty*, AI Game Programming Wisdom, Editado por Ravin Steve, Charles River Media, 2002, pp. 345-357, ISBN: 978-158-45-0077-3

[UNRT08] Unreal Technology, Epic Games, Inc.
<http://www.unrealtechnology.com/>
Consultado en Noviembre 2008

[VELA08] Velasco-Santos Perla, Laureano-Cruces Ana Lilia, Mora-Torres Martha, Sánchez-Guerrero Lourdes. *La Importancia del Diseño de una Interfaz en el Proceso de Enseñanza-Aprendizaje* XXI Congreso Nacional y VII Congreso Internacional de Informática y Computación de la ANIEI. Monterrey 1-3 de octubre del 2008.

[WALK08] Walker Mitch, *What's New in XNA Game Studio 3.0*, Gamefest: Microsoft Game Technology Conference, 2008

[WATA92] Watt Alan, Watt Mark, *Advanced Animation and Rendering Techniques Theory and Practice*, Addison-Wesley, 1992, ISBN: 978-020-15-4412-1

[WILI99] Wilson Ian, *The Artificial Emotion Engine, Driving Emotional Behavior*, Artificial-Emotion.Com, November 1999

[WILR99] Wilson R.A., Keil F.C. *The MIT Encyclopedia of the Cognitive Sciences*, MIT Press, Cambridge/Boston/London, 1999, 978-0-262-23200-5

[WUKV67] Wukmir V.J., *Emoción y Sufrimiento* Endoantropología elemental., 1967.

<http://www.biopsychology.org/wukmir/eys/eys.htm>
Consultado en Marzo 2008

[XCCO09] XNA Creators Club Online, Microsoft Corporation, 2009
<http://creators.xna.com/en-US/>
Consultado en Enero 2008

[XEOD08] XEODesign, Inc.
<http://www.xeodesign.com>
Consultado en Noviembre 2008