

Contenido

Contenido

Índice de figuras

1. Introducción

2. Antecedentes

- 2.1. 2.1 La Creatividad
- 2.2. Creatividad, Inteligencia e Imaginación
 - 2.2.1. El Proceso Creativo
 - 2.2.2. Las Características del Proceso Creativo
 - 2.2.3. Los Bloqueos durante el proceso Creativo
- 2.3. Creatividad Computacional
 - 2.3.1. Técnicas usadas en la Creatividad Computacional
- 2.4. Ejemplos de Creatividad Computacional
 - 2.4.1. Bomba de Imágenes
 - 2.4.2. Arte Evolutivo
 - 2.4.3. Neuro-Arte Evolutivo (NevAr: Neural Evolutionary Art)
 - 2.4.4. AARON
 - 2.4.5. Mistrel
 - 2.4.6. MEXICA
 - 2.4.6.1. Cómo Escribimos
 - 2.4.6.2. Aspectos Generales de MEXICA
- 2.5. IA en los Videojuegos
 - 2.5.1. Dungeon Beeper
 - 2.5.2. Field of battle
 - 2.5.3. SWAT
 - 2.5.3.1. IA Estratégica
 - 2.5.3.2. IA Táctica

3. Adaptación del modelo Ensimismamiento-Reflexión

- 3.1. Especificación de las Reglas del Juego
- 3.2. Definición de la Interacción Jugador-Ambiente de Juego
- 3.3. Los Agentes Racionales
- 3.4. El manejo de mensajes Emitidos en el Ambiente de Juego
- 3.5. El Tablero de Juego
- 3.6. El esquema General de la Implementación
- 3.7. Descripción Detallada de la Adaptación del Modelo Ensimismamiento-Reflexión
 - 3.7.1. La Estructura Maestra
 - 3.7.1.1. La Sección de Estrategia

- 3.7.1.2. La Sección del Objetivo Final
- 3.7.1.3. La Sección de Objetivos Particulares
- 3.7.1.4. La Sección del Plan de Juego
- 3.7.1.5. La Sección de Asignación de Acciones
- 3.8. Etapas de la Implementación de la Adaptación del Modelo Ensimismamiento-Reflexión
 - 3.8.1. Etapa 1: Obtención del Experiencia Previa
 - 3.8.2. Etapa 2: Procesamiento de la Experiencia Previa
 - 3.8.3. Etapa 3: Implementación de la Adaptación del Modelo del Proceso Creativo
 - 3.8.3.1. Iniciando el Juego
 - 3.8.3.1.1. Inicio Completamente Aleatorio
 - 3.8.3.1.2. Inicio Aleatorio sobre la Representación Abstracta
- 3.9. Esquema General del Funcionamiento de la Adaptación del Modelo Ensimismamiento-Reflexión
- 3.10. Obtención de los Datos del ambiente de Juego y Generación de la Semilla Inicial
- 3.11. El Estado Ensimismamiento
 - 3.11.1. Definición de las tolerancias de comparación entre los datos comparados
 - 3.11.2. Eliminación de datos (los filtros)
- 3.12. El Estado Reflexión
- 3.13. La Asignación de Acciones
- 3.14. Las Acciones Cognitivas
- 3.15. Los Estancamientos

4. Cómo Funciona la Implementación de la Adaptación del Modelo Ensimismamiento-Reflexión

- 4.1. Obtención de la Experiencia Previa
- 4.2. Preprocesamiento de la Experiencia Previa
- 4.3. Construcción de la Representación Abstracta del Conocimiento
- 4.4. Cómo Funciona la Adaptación del Modelo Ensimismamiento-Reflexión
 - 4.4.1. Generando la Semilla Inicial
 - 4.4.2. Determinando los Objetivos Particulares
 - 4.4.3. Creando el Plan de Juego
 - 4.4.4. Un Ejemplo de ejecución
 - 4.4.4.1. Generación de la Semilla Inicial
 - 4.4.4.2. Iteraciones entre los Estados Ensimismamiento-Reflexión
 - 4.4.4.3. Cómo Intervienen las Acciones Cognitivas
 - 4.4.4.4. Los eventos en el ambiente de Juego como Banderas para la Intervención de la Implementación del Modelo Ensimismamiento-Reflexión.

5. Resultados y Evaluación

- 5.1. Evaluación
- 5.2. Resultados
 - 5.2.1. Elección de la semilla inicial
 - 5.2.2. Intervención de la Implementación de la Adaptación del Modelo Ensimismamiento-Reflexión para la generación del Plan de Juego
 - 5.2.3. Asignación de Acciones

- 5.2.4. Reacción a los eventos en el ambiente de Juego
 - 5.2.4.1. Procesando el evento “Aparece Tornado”
 - 5.2.4.2. Procesando el evento “Aparece Tarjeta”
- 5.2.5. Ensimismamiento
- 5.2.6. Reflexión
- 5.2.7. Asignación de Acciones

6. Conclusiones

- 6.1. Respuestas a las preguntas introductorias
 - 6.1.1. ¿Cómo representar el conocimiento?
 - 6.1.1.1. Conclusión 1
 - 6.1.1.2. Conclusión 2
 - 6.1.2. ¿Es posible la adaptación del modelo Creativo Ensimismamiento-Reflexión que originalmente fue implementado en un contexto literario a un Videojuego de Computadora?
- 6.2. Aportaciones de esta Investigación
 - 6.2.1. Aportaciones
- 6.3. Propuestas para trabajos Futuros
- 6.4. Conclusión general

7. Apéndices

Índice de Figuras

2. Antecedentes.

- 2.1. Arte Generado por Bomba de Imágenes (Algoritmos genéticos)
- 2.2. Ejemplo de Arte Evolutivo de William Latham (Programación Genética)
- 2.3. Algoritmo Creativo usado en NevAr
- 2.4. Arte Producido por NevAr.
- 2.5. Pantalla del Videojuego Dungeon Beeper
- 2.6. Pantalla del Videojuego Field of Battle
- 2.7. Pantalla del Juego SWAT

3. Adaptación del modelo Ensimismamiento-Reflexión

- 3.1. Interfaz del Programa “GeneraMensajes.exe”
- 3.2. Interfaz del Videojuego *Guns*
- 3.3. Esquema General de la Adaptación del Modelo Ensimismamiento-Reflexión al Videojuego de Estrategia
- 3.4. Tabla de resumen de un ejemplo de la especificación de los Parámetros que definen una Estrategia de juego en el Videojuego *Guns*
- 3.5. Composición Interna de la Estructura Maestra
- 3.6. Representación de la Estructura Maestra en Código de Computadora
- 3.7. Esquema del Procesamiento de la Experiencia Previa
- 3.8. Archivo de Experiencia Previa
- 3.9. Interfaz para la captura de un comentario en Tiempo de Juego
- 3.10. Interfaz del programa “ProcesaExperiencia”
- 3.11. Ejemplo de Registro procesado por el programa “ProcesaExperiencia”
- 3.12. Archivo de Representación Abstracta
- 3.13. Esquema General del Funcionamiento de la Adaptación del modelo Ensimismamiento-Reflexión
- 3.14. Esquema General del Funcionamiento del Modelo Ensimismamiento-Reflexión
- 3.15. Implementación de la generación Aleatoria de la Semilla Inicial para el Funcionamiento del Modelo Ensimismamiento-Reflexión.
- 3.16. Implementación de la Generación de la Representación Abstracta
- 3.17. Ejemplo de la captura de Datos en tiempo de Juego
- 3.18. Abstracción de la Información como Estructura Asociativa
- 3.19. Ejemplo de una Estructura Asociativa
- 3.20. Ejemplo de un Átomo de Comparación
- 3.21. Seguimiento de la Comparación y Transformación entre Estructura Asociativa y átomos de la Representación Abstracta del Conocimiento
- 3.22. Ejemplo de la especificación de los Parámetros de la Estrategia de Juego.

4. Cómo Funciona la Implementación de la Adaptación del Modelo Ensimismamiento-Reflexión

- 4.1. Colección de Datos Guardados la hacer una Tirada en el Videojuego *Guns*
- 4.2. Colección de Datos Capturados durante una Tirada en el Videojuego *Guns* procesada por el Programa “ProcesaExperiencia”
- 4.3. Ejemplo de una Estructura Átomo

- 4.4. Estandarización de un Comentario Agregado por el Usuario
- 4.5. Representación de un Comentario agregado por el Usuario en el Tiempo de Juego en la Estructura Átomo.
- 4.6. Ejemplo del resultado del Proceso que construye las Estructuras “Objetivos Particulares” y “Plan de Juego”
- 4.7. Resultado de la Selección Aleatoria de la Semilla Inicial
- 4.8. Resultado de la Ejecución de un primer Ciclo de la Implementación de la Adaptación del Modelo
- 4.9. Resultado de la Ejecución de un Segundo ciclo de la Implementación de la Adaptación del Modelo
- 4.10. Resultado de la Iteración entre Ciclos de la Implementación de la Adaptación del Modelo
- 4.11. Resultado Final de la Iteración entre Ciclos entre Ciclos de la Implementación de la Adaptación del Modelo

5. Resultados y Evaluación

- 5.1. Átomo seleccionado aleatoriamente para la especificación de la semilla inicial.
- 5.2. Resumen de los parámetros que definen la Estrategia de juego.
- 5.3. Resultado de la generación del Plan de Juego
- 5.4. Resultado de la asignación de acciones
- 5.5. Equipo controlado por la IA ejecutando la acción IR_POR_METAL2
- 5.6. Lanzamiento de los eventos “Tornado” y “Tarjeta”
- 5.7. Estructuras Asociativas de los tres agentes al momento de lanzarse el evento “Tornado”
- 5.8. Primear transformación para una EA para el evento Tornado
- 5.9. EA para los tres agentes del equipo controlado por la IA para el evento “Tarjeta”
- 5.10. Resultado obtenido de una primera búsqueda durante el estado de Ensimismamiento.
- 5.11. Resultado del filtro de Información en una EA debida al evento “Tarjeta”
- 5.12. Conjunto de posibles acciones siguiente obtenidas del estado de Ensimismamiento
- 5.13. Estructura de asignación de tareas resultante después del estado Reflexión

6. Conclusiones

- 6.1. Esquema general de un agente implementado en este proyecto de investigación.

Capítulo 1

Introducción

Los hombres nos hemos denominado a nosotros mismo como “*Homo sapiens*”. Término latino que alude a la capacidad de razonamiento y que pone de relieve lo importante que nos resultan nuestras facultades y capacidades mentales. Por lo que durante miles de años se ha tratado de entender como una aparentemente simple masa (cerebro) puede percibir, entender, predecir y manipular un mundo mucho más complicado y grande que ella misma. El campo de la **Inteligencia Artificial** o IA, va mucho más allá, pues no sólo intenta comprender, sino también se esfuerza por construir entidades inteligentes.

En la actualidad la IA abarca una gran variedad de subcampos que van desde áreas de propósito general, como el aprendizaje y la percepción a otras más específicas como el ajedrez, los juegos electrónicos, la demostración de teoremas matemáticos, la escritura de poesía y el diagnóstico de enfermedades. Por todo esto se puede decir que la IA sintetiza y automatiza tareas intelectuales y es, por lo tanto, potencialmente relevante para cualquier ámbito de la actividad intelectual humana. En este sentido, es un campo genuinamente universal.

El presente trabajo está orientado a probar el alcance de un modelo cognitivo denominado Ensimismamiento-Reflexión el cual está basado en las conclusiones derivadas del estudio del ejercicio intelectual de un escritor para generar material (en su caso específico, literatura) creativo e interesante. Este modelo ha sido adaptado a un Videojuego de estrategia para computadora el cual tiene por propósito generar en

tiempo de juego una estrategia tal que al enfrentarse a un jugador humano intente ganar la partida.

De modo que en el desarrollo de este trabajo de tesis tiene como objetivo contestar preguntas obligadas que tienen que ver con la especificación del modelo usado:

- ¿Cómo se representa el conocimiento? Es decir, ¿Es posible representar e interpretar la información contenida en un ambiente dinámico? Y si es así... ¿Cuánta de esa información es relevante para poder tener un desempeño exitoso por parte de un ente que se desarrolla en ese ambiente? ¿Podrá la IA filtrar, abstraer y usar esa información para tomar decisiones?
- ¿Es posible la adaptación del modelo creativo Ensimismamiento-Reflexión que originalmente fue implementado en un contexto literario a un videojuego de computadora?
- Si la respuesta a la última pregunta es afirmativa entonces, nos interesa saber:
 - La implementación y adaptación del modelo... ¿Funciona? Es decir, ¿Es capaz de ganarle a un jugador humano?
 - En el caso de encontrarse con dificultades al ser implementado el modelo entonces... ¿En qué reside esa dificultad? ¿Es superable?

Para contestar a estas preguntas, como ya se a dicho, se ha creado un Videojuego de estrategia para computadora el cual hace uso de la implementación de la adaptación del modelo del procedo creativo Ensimismamiento-Reflexión desarrollado por el Dr. Rafael Pérez y Pérez [7] para la generación de una estrategia de juego que lleve a la posible victoria y que haga uso de las características de los miembros del equipo que es dirigido por esta implementación y de un proceso de constante evaluación del ambiente en un momento determinado.

La intervención de la implementación está directamente influenciada por la percepción de señales (eventos) y es guiada por dos clases de acciones:

1) Acciones cognitivas (procesos de evaluación y propiamente los estados de Ensimismamiento y reflexión).

2) Acciones de interacción (la toma de decisiones que se reflejan en la reacción del equipo con el ambiente y que son percibidas en el ambiente grafico como comportamiento de los agentes).

De modo que el modelo recibe como entrada un estado que es descrito por la serie de datos numéricos, boléanos y cadenas de texto y tiene como salida cadenas de texto que representan acciones y que finalmente se traducen en reacciones en el ambiente.

Tiene que decirse que este trabajo, así como se ha descrito, abarca varios campos de la IA, los cuales son esenciales para hacer que un sistema inteligente pueda funcionar en un sistema dinámico, estos son:

1. Percepción de señales.
2. Representación del conocimiento.
3. Razonamiento.
4. Interacción con el ambiente.

Todos estos campos han sido tratados e implementados en software y son considerados en este trabajo como necesarios para la implementación de la adaptación de este modelo creativo a un ambiente dinámico.

Tiene que mencionarse además que se tuvo que desarrollar programas auxiliares los cuales ayudaron en el análisis de la dinámica del juego y en la obtención de la información relevante, así como para el procesamiento y filtrado de la misma. Todos estos se explicarán con detalle a lo largo de este documento.

Capítulo 2

Antecedentes

Evidentemente, la investigación en Inteligencia Artificial (IA) se ha dividido en dos vertientes bien identificadas: 1) La automatización de tareas que requieren de racionalidad¹ y 2) El estudio del funcionamiento de la mente humana durante el ejercicio racional y su emulación en máquinas. Ambas orientaciones han contribuido en su momento y a su manera al progreso tecnológico, ya sea independientes una de la otra o trabajando como equipo complementándose entre ellas.

Mientras que la primera vertiente se ha ocupado de apoyar a expertos en distintas áreas del conocimiento y hasta resolver problemas más rápido y en su medida mejor que los humanos, la segunda se ha encargado de emular el proceso del ejercicio del razonamiento. Esta segunda vertiente de la IA propone modelos cognitivos que pretenden explicar lo que sucede en la mente cuando se razona, se inventa, se crea o gesta una nueva y novedosa idea teniendo principalmente como variable de evaluación la racionalidad de los resultados obtenidos.

La idea de poder imitar correctamente la manera y el orden en que suceden los eventos dentro de una mente humana es muy tentadora. El simple hecho de poder automatizar procesos inventivos, deductivos y analíticos simplemente le daría capacidades ilimitadas al progreso tecnológico y científico. Sin embargo, lo que sucede en un cerebro es demasiado complicado y estamos muy lejos de entenderlo por entero.

¹ Específicamente en este contexto, se entiende por racionalidad como la facultad intelectual de juzgar los resultados obtenidos como razonables o buenos, los que se acercan lo mejor posible a lo deseado tomando en cuenta las circunstancias predominantes y el objetivo planteado.

Por lo anterior queda claro que la simulación está –hasta ahora, muy lejos de poderse lograr, por lo que se ha recurrido a la emulación del proceso cognitivo, lo que ha permitido el nacimiento de áreas como la Realidad Virtual, la Computación Evolutiva y la Computación Creativa.

El presente trabajo esta orientado hacia la Creatividad Computacional, y presenta la adaptación de un modelo del proceso creativo (Ensimismamiento-Reflexión, Pérez y Pérez [7]) a la generación de estrategias de juego a un videojuego de computadora. Este modelo representa un punto intermedio entre las dos vertientes de la inteligencia artificial, es decir, la solución automática de problemas y la emulación del proceso creativo.

Como se verá una de las mayores contribuciones de este trabajo es mostrar el alcance del modelo usado, adaptado e implementado en cuanto a la automatización de tareas creativas se refiere. Y su justificación es apoyada en gran medida por uno de los objetivos básicos en la creación de los videojuegos, a saber, la simulación de conductas humanas, ya sean individuales o en equipo y el enfrentamiento entre la implementación de estas mediante lenguajes de programación y las naturales e intrínsecas de un jugador humano.

Debe ser aclarado que el presente trabajo se desliga completamente del ámbito Psicológico al no intentar analizar e implementar correctamente el proceso creativo puramente humano, sino se enfoca exclusivamente a la evaluación de la funcionalidad, racionalidad y alcance del modelo usado en el ámbito de la IA usada en videojuegos.

Así pues, en este capítulo se define la Creatividad en el modo en que se usa en este trabajo, se hace una breve revisión de la investigación actual en cuanto a la automatización del proceso creativo en computadoras incluyendo y dando especial atención al modelo que representa la base de este trabajo, así como un breve análisis también del uso y particular implementación de la Inteligencia Artificial en el ámbito de los videojuegos en cuyos casos específicos se trata de obtener como resultado final, mediante ingeniosas soluciones, comportamientos similares a las que tiene un ser humano al jugar un videojuego de estrategia.

2.1 La Creatividad

El concepto de creatividad no se ha fijado con certeza. Existen muchas percepciones y opiniones en cuanto lo que es. Pero entre tantas opiniones existe un

punto común: “La creatividad es una cualidad, virtud o don que se puede apreciar en diferentes niveles y en todo ámbito”. La creatividad es un potencial que se puede entrenar, desarrollar o evolucionar. No se enseña, sino que se desbloquea. En el caso particular del ser humano, si el entorno local (físico y sociocultural) es capaz de estimular, motivar o alentar, entonces la creatividad florece. Según los psicólogos la creatividad no aparece súbitamente ni por generación espontánea. Tampoco es un accidente, sino un proceso continuo: se agregan constantemente pequeñas ideas nuevas a ideas previamente existentes. Básicamente, la creación artificial –como la natural- se compone de dos procesos computacionales: uno “generador” y otro “evaluador”.

Pero aunque bien es cierto que se desconoce aun la exactitud con la que tiene lugar el proceso creativo de la mente humana, P.J Bentley afirma en su artículo *Is Evolution Creative?* [1] que “La creatividad puede ser reconocida por sus manifestaciones”. Basándose en ideas como esta, la simulación de los procesos mentales ha sido campo casi exclusivo de la IA y ha tratado de imitarla usando como herramienta principal la computadora.

Varias técnicas de Inteligencia Artificial ya han demostrado que es factible modelar actividades creativas en muchos campos: artes (música, literatura), ciencias (física, matemática), tecnologías (ingeniería, arquitectura), juegos (ajedrez, damas), etc. (Moriello, 2001).

Como ya se ha mencionado, para la creatividad no existe una única definición. Pero se podría decir que la creatividad es la capacidad de unir, juntar, asociar, conectar, integrar o combinar diferentes ideas ya existentes –y previamente no relacionadas de manera no habitual, inesperada, sorpresiva, impredecible e innovadora, a fin de producir nuevas ideas más complejas y potentes adaptadas a los nuevos propósitos. No obstante, es preciso que sean útiles, pertinentes, razonables y/o viables; es decir, se deben descartar todas aquellas ideas descabelladas, inalcanzables o absurdas. De allí que la creatividad es un concepto parcialmente subjetivo y relativo a un contexto sociocultural: lo que en una cultura puede ser valioso, útil o beneficioso en otra puede no serlo.

En el caso del ser humano, la creatividad no es algo exclusivo de algunas personas sino que es inherente a toda la especie. Aunque, conviene aclarar, no todos son igualmente creativos: algunos lo son mucho más que otros e incluso unos pocos son extraordinarios. La creatividad es un potencial, que se puede entrenar, desarrollar,

evolucionar. No se enseña sino que se desbloquea. Si el entorno local (físico y sociocultural) es capaz de estimular, motivar o alentar, entonces la creatividad florece.

2.2 Creatividad, Inteligencia e Imaginación

La creatividad no necesariamente va aparejada con la inteligencia (se puede ser mucho más creativo que inteligente y viceversa), pero depende de ella. También está relacionada con la imaginación: crear es poner de manifiesto aquello que se imagina. En efecto, todo aquello que fue construido por el hombre –desde los primeros objetos de piedra hasta las complejas estaciones espaciales - fue concebido originariamente por la imaginación. Por último, depende también del conocimiento y de la experiencia, aunque no se debe caer en sus extremos: si bien es muy útil contar con un bagaje mínimo, el exceso de conocimientos y de experiencias puede restringir la creatividad.

Aparentemente para ser creativo son condiciones esenciales la motivación y la colaboración entre las personas; mientras que son aspectos importantes o necesarios el aprendizaje y la interacción con el entorno (físico y social). En términos generales, las personas creativas no sólo dependen del proceso cognitivo sino que también dan importancia a aspectos afectivos de su personalidad para la solución de los problemas que se les presenta.

2. 2. 1 El proceso Creativo

La creatividad no aparece súbitamente, por generación espontánea. No es un accidente, sino un proceso continuo: se agregan constantemente pequeñas ideas nuevas a ideas previamente existentes. Los grandes saltos son muy poco frecuentes.

El proceso creativo consta, básicamente, de dos etapas: una “generadora” y otra “evaluadora”. Muchos autores lo descomponen, a su vez, en cuatro “momentos” distintos (dos conscientes, separados por dos inconscientes) (Penagos y Aluni, 2000):

- *Preparación*: se selecciona y recolecta la mayor cantidad de información posible y viable sobre el problema en cuestión, incluyendo lecturas, indagaciones, averiguaciones, consultas, etc.

- *Incubación*: se deja “decantar” el tema. Se trata de un período de aparente inactividad, pero se produce un procesamiento subconsciente de la información recolectada.

- *Iluminación*: se produce el “relámpago creativo”, la inspiración propiamente dicha, el momento en que “caen todas las fichas”. La solución surge abrupta y repentinamente en la mente.

- *Verificación*: es el trabajo de evaluación –racional y metódica- a fin de determinar la validez objetiva y/o práctica de la solución obtenida en la etapa de iluminación.

De este modo, la preparación y la incubación corresponden a la etapa generadora y la iluminación y la verificación a la evaluadora, respectivamente. Otros autores agregan otros dos “momentos” más (al inicio y al final del proceso):

- *Cuestionamiento*: se percibe algo como problema y se inicia una reflexión profunda a fin de delimitarlo con claridad y

- *Comunicación*: es la tarea de dar a conocer lo nuevo, de hacer una adecuada difusión de lo creado.

2.2.2 Las características del proceso creativo

Las características más comúnmente aceptadas de la creatividad, en la gran mayoría de las personas, son (Penagos y Aluni, 2000):

- Fluidez: es la facilidad para generar un elevado número de ideas. Obviamente cuanto mayor es su número, más probable resulta encontrar una alternativa o solución útil y beneficiosa.

- Flexibilidad: es la habilidad para modificar, transformar, replantear o reinterpretar –de muy diversas formas– ideas, explorando líneas diferentes.

- Originalidad: es lo que define a la idea como algo único, nuevo, inusual, novedoso, diferente o inédito. Significa apartarse de lo obvio, de lo habitual, romper la rutina, dar un salto desde lo conocido.

- Elaboración: es el nivel de detalle, desarrollo o complejidad de las ideas creativas.

2.2.3 Los bloqueos durante el proceso creativo

En la gran mayoría de las personas, existen cinco tipos de bloqueos que inhiben la creatividad (Penagos y Aluni, 2000):

- *Bloqueos perceptuales*: son las limitaciones, constricciones y/o restricciones para percibir claramente cuál es el problema, para enfocarlo en su totalidad, para ver relaciones remotas, para investigar lo obvio, etc.

- *Bloqueos cognitivos*: son las limitaciones, constricciones y/o restricciones propias del conservadurismo y de los conocimientos y experiencias adquiridas. Es el no uso de la imaginación, el apego a la lógica, la excesiva familiaridad con un tema, el abandono de los sueños, etc.

- *Bloqueos afectivos*: son las limitaciones, constricciones y/o restricciones a la propia libertad para investigar y manipular ideas. Son las inseguridades internas, el temor a lo desconocido, el miedo al ridículo, los prejuicios propios, la baja autoestima, etc.

- *Bloqueos conativos*: son las limitaciones, constricciones y/o restricciones que produce la falta de curiosidad, de motivación, de deseo, de actitud, de voluntad, etc.

- *Bloqueos socioculturales*: son las limitaciones, constricciones y/o restricciones que imponen ciertos patrones, normas, paradigmas y marcos de referencias sociales y culturales. Son las antiguas tradiciones, los convencionalismos, el miedo de equivocarse, la forma en que uno ha sido educado, etc.

2.3 Creatividad computacional

Básicamente, la creación artificial –como la natural – se compone de dos procesos computacionales: uno “generador” y otro “evaluador”. En cuanto al primer proceso, es fácil para la máquina generar muchas de las soluciones posibles a un determinado problema.

Es así que, en muchas ocasiones y dado que funciona a toda hora (y durante días), encuentra resultados inesperados, soluciones originales, que el ser humano posiblemente habría pasado por alto. Lo que ocurre es que sólo es capaz de combinar “ideas” previamente conocidas, no puede realizar asociaciones inéditas, transgredir, “traspasar los límites” o “romper las reglas” como hacen los grandes creativos.

En cuanto al segundo proceso, es difícil evitar que la máquina haga cosas inútiles. Por el momento, no posee la capacidad de evaluar y, por lo tanto, de escoger únicamente la solución más adecuada. No obstante, se puede implementar un “sistema creador”, compuesto por una máquina y por un ser humano, en donde la computadora

genere abundante material re combinado y el hombre la guíe por el camino más interesante o la máquina efectuara los procesos inferiores de una obra y el ser humano continuara el trabajo en un nivel superior.

2.3.1 Técnicas utilizadas en la creatividad computacional

Varias técnicas de Inteligencia Artificial ya han demostrado que es factible modelar actividades creativas en muchos campos: artes (música, literatura), ciencias (física, matemática), tecnologías (ingeniería, arquitectura), juegos (ajedrez, damas), etc. Algunas de ellas se describen sintéticamente a continuación:

- Sistemas Expertos: emulan artificialmente ciertos aspectos del razonamiento de un especialista humano en un ámbito restringido y limitado de conocimiento. Su característica principal es que se basan en reglas; es decir, contienen un juego predefinido de conocimientos que se utiliza para tomar sus decisiones. Resuelven con facilidad ciertos problemas complejos que sólo puede llevarse a cabo por personas expertas intensamente entrenadas (Moriello, 2001):

- Razonamiento Basado en Casos: aprovecha la experiencia adquirida en el pasado para resolver el problema actual, a través de su gran base de conocimientos con ejemplos de casos ya resueltos (históricos). A la hora de resolver un nuevo caso, el sistema busca en su memoria y recupera aquel que más se le asemeje adaptándolo al problema actual. Esta nueva solución, por lo tanto, pasa a engrosar la base de conocimientos actualizada. Al incorporar permanentemente nuevos casos a su memoria, el sistema va adquiriendo más “experiencia” con el tiempo, la cual le permite ir desempeñándose cada vez mejor. Se trata, en definitiva, de un aprendizaje por.

- Redes Neuronales Artificiales: inspiradas en el cerebro humano, están compuestas por una multitud de procesadores paralelos interconectados, cada uno de los cuales es capaz de efectuar sólo un pequeño número de operaciones simples y transmitir sus resultados a sus vecinas. A las redes neuronales no se les “inculca” ningún tipo de regla, sino que son capaces de aprender a reconocer patrones, a partir de un proceso de entrenamiento basado en el análisis automático y sistemático de una suficiente cantidad de ejemplos diferentes. Son hábiles para manipular datos imprecisos, incompletos, con ruidos y hasta compuestos de ejemplos contradictorios.

- Algoritmos Genéticos: son métodos adaptativos de búsqueda que se basan en los mecanismos de evolución biológica. En ellos se codifica cada una de las posibles soluciones a un problema dado en forma de cadenas de caracteres de longitud fija

llamados “genes”. Se genera –normalmente al azar – una “población” inicial de prueba (un conjunto de posibles soluciones con ligeras variaciones entre ellas), a la cual se evalúa posteriormente según un criterio de desempeño fijado con anterioridad (la “función de adecuación” o fitness). En cada ciclo (cada “generación”) se eligen las soluciones que más se aproximan al objetivo buscado, descartando el resto de las soluciones. Aquellas seleccionadas (“las más aptas”) se combinan (“reproducen”) entre sí para producir nuevas soluciones (su “descendencia”), permitiendo –de vez en cuando– introducir alguna modificación al azar (una “mutación”) durante la reproducción. El ciclo se repite muchas veces, quizás cientos, hasta llegar a aquella considerada aceptable.

- Sistemas Multiagentes: es un conjunto de entidades relativamente autónomas e inteligentes que cooperan entre sí para desarrollar una tarea o resolver un problema. Se trata de comunidades de agentes, cuyas propiedades no pueden derivarse únicamente de las de sus partes constitutivas. Es posible incrementar su capacidad aumentando el número de bases de conocimientos especializadas.

La técnica de IA que más importancia ha tenido en el ámbito de la creatividad han sido los algoritmos genéticos, los cuales se han aplicado en la generación de arte, diseño y solución de problemas.

2.4 Ejemplos de Creatividad Computacional

Es muy variado el trabajo que se ha hecho en el ámbito de la creatividad computacional. Se han abarcado campos que van desde lo artístico, el diseño, la ingeniería y la asistencia a expertos como generador de nuevas alternativas e ideas. Sin embargo, en esta sección se revisa algo del trabajo relacionado a la creatividad computacional. Así mismo se hace la revisión de los aspectos generales del modelo usado y de algunas técnicas de IA usadas en el ámbito de los videojuegos.

2.4.1 Bomba de Imágenes

Scott Draves ha usado los algoritmos genéticos para producir lo que el llama como gráficos orgánicos animados o “Bombas” los cuales responden a la activación de teclas produciendo lo que el mismo califica como “Música visual”, esto es, una cadena de video que es fluido, texturizado, rítmico y animado. Acott Draves afirma: “La característica esencial de la vida es la expansión creativa, cuando esta es alimentada se

origina una descompresión de información que debe y es procesado y reutilizada, a diferencia de los comportamientos estáticos, repetitivos, aleatorios o predecibles, la vida tiene la habilidad de seguir creando nuevas cosas”.

Usando grandes secuencias geométricas de formas básicas y simétricas así como secuencias rítmicas controladas y administradas por Algoritmos genéticos logra producir lo que se ha denominado como creatividad Evolutiva Visual.

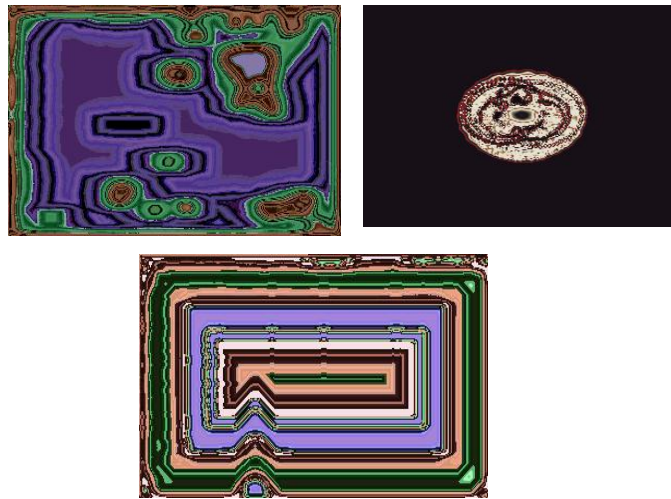


Fig. 2.1 Arte generado por Bomba de Imágenes (Algoritmos Genéticos)

2.4.2 Arte evolutivo

El trabajo de William Latham es la única fusión de arte y tecnología que ha sido completamente producida usando “Programas Mutantes” (Desarrollado por él mismo) en los que usa programación genética. Latham comenzó con estructuras simples en 3D las trabaja como lo haría un jardinero o un agricultor, esto es, seleccionando, cruzando y produciendo nuevos individuos para obtener nuevos individuos mutantes complejos. El trabajo de Latham es considerado como el único trabajo Darwiniano manejado por las estéticas humanas. El la califica como un parodia de la manipulación cruel del hombre del mundo natural por la ingeniería genética y el desmedido e inconciente cultivo moderno, de lo cual a obtenido resultados fascinantes. Este trabajo ha sido catalogado como “Sumamente hermoso”, “Profundamente siniestro”, “Paranóico”, “Computación natural”.

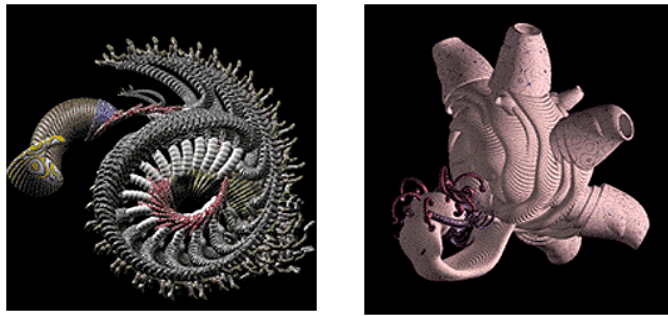


Fig. 2.2 Ejemplo del Arte Evolutivo de William Latham (*Programación Genética*)

2.4.3 Neuro-Arte Evolutivo (NEvAr: Neural Evolutionary Art)

El neuroarte Evolutivo es un trabajo desarrollado por el Grupo de sistemas creativos quienes se dedican a las áreas como la música, Generación de imágenes, Diseño, Creatividad e IA. Esta implementación representa la primera aplicación hecha por el grupo en una aproximación al arte evolutivo orientado a la generación de imágenes.

Está hecho en C++ y corre bajo Windows y se le considera como una aplicación de Programación Genética (PG) inspirada en el trabajo de K.Sims y R. Dawkins. NEvAr es guiado por el usuario quien contribuye con valores de la evaluación del desempeño del programa y decide cuando terminar.

A groso modo lo que hace NEvAr es descartar poblaciones anteriores a la actual para evaluación actual, sin embargo las conserva para hacer cruza entre las nuevas poblaciones generadas. Está creado en una arquitectura multidocumentos que permite trabajar con varios experimentos a la vez y la migración entre proyectos.

El “algoritmo creativo” que se usa en su implementación se muestra en la siguiente figura:

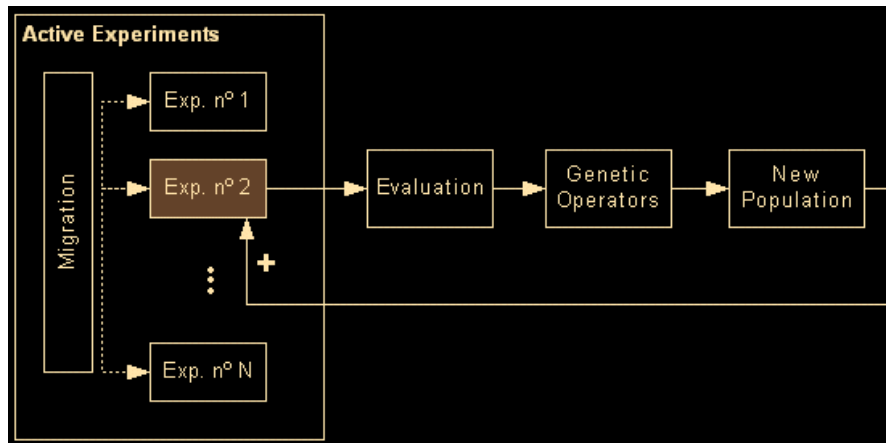


Fig. 2.3 Algoritmo creativo usado en NevAr

NEvAr está basado en dos aspectos: 1) Utilidad y 2)Autonomía, Para mejorar la Utilidad sus creadores diseñaron una interfaz robusta (estable) y rápida. Pero en cuanto la autonomía es parcial debido a que el estado de evaluación es ejecutada por el usuario. Actualmente trabajan en la implementación en el mismo sitema de Redes Neuronales para filtrar algunas de las imágenes que no correspondan a los gustos del usuario actual.

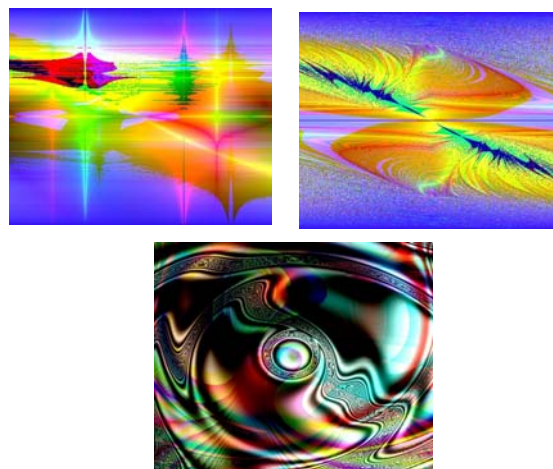


Fig.2.4 Arte producido por NevAr

2.4.4 AARON

Harold Cohen tenía interés en explorar la forma en que los humanos crean y leen representaciones como dibujos, pinturas, siluetas, etc. Creía que si él podía enseñar a dibujar a una computadora, entonces podría entender mejor este proceso. Así, en 1971 comenzó a escribir un programa capaz de crear dibujos originales.

Basado en su propia experiencia como artista, creó Aaron, un sistema experto que intenta modelar el comportamiento de un artista a diferentes niveles. En un alto nivel el programa toma decisiones acerca de la organización y composición de sus imágenes. Mientras que, en un nivel bajo, genera las instrucciones para dibujar las líneas y siluetas que conforman un dibujo. A lo largo de los años, los dibujos de Aaron han evolucionado desde simples líneas a dibujos coloreados de figuras humanas y vegetales. Sin embargo se sabe muy poco acerca de cómo funciona, por lo que la afirmación de que Aaron es un artista es muy controvertida. En particular, todo parece indicar que Aaron contiene especificaciones sobre las dimensiones de la figura humana y diferentes posiciones en que se puede encontrar que usa aleatoriamente en sus dibujos, sin tener ninguna intención de transmitir o reflejar alguna idea, tema o significado. Más aún, es muy probable que su conocimiento del ser humano, sus actividades, sentimientos, etc. sea muy vago, si no es que nulo.

2.4.5 Minstrel

Minstrel es un programa creado por Scott Turner [1] que escribe cuentos cortos sobre el Rey Arturo y sus Caballeros de la mesa redonda. Es un sistema de resolución de problemas basado en casos, en el que los casos pasados son almacenados en una memoria episódica. En Minstrel todos los elementos que conforman un cuento son representados como esquemas: esquemas de autor y esquemas de personaje. Los esquemas de autor son usados por el sistema para satisfacer las metas del escritor, como el desarrollo de un tema y la inclusión de suspenso en la historia. Las metas del escritor tienen asociado un conjunto de instrucciones para ser alcanzadas. Estas instrucciones indican cómo crear escenas de venganza, decepción, etc. Los esquemas de personaje son usados por el sistema para representar eventos en un cuento. Ejemplos de esquemas de personaje son las metas de los personajes (p.e. satisfacer hambre, encontrar un amor), representaciones de humanos, objetos físicos y emociones. Los esquemas de caracteres pueden asociarse para establecer relaciones entre ellos. De esta manera es posible construir escenas elaboradas. Minstrel desarrolla cuentos acerca de 6 temas predefinidos y los cuentos en Minstrel tienen una estructura definida. Para desarrollar un cuento, Minstrel ejecuta 2 procesos principales: el proceso de planeación, que controla las metas del escritor, y un proceso de resolución de problemas, enfocado en lograr las metas. Además, como parte del proceso de desarrollo de un cuento, cada vez que una escena es creada el sistema revisa la

consistencia de la historia del cuento y busca oportunidades para incluir elementos dramáticos.

En Minstrel, escribir un cuento consiste en instanciar todos los esquemas que conforman un tema. Cuando el sistema no puede encontrar eventos en su memoria episódica para instanciar el tema, o los eventos disponibles fueron usados más de una vez en cuentos anteriores, se emplea un conjunto de heurísticas para crear escenas nuevas.

2.4.6 MEXICA

En 1996 Mike Sharples [34] publica una serie de observaciones sobre la creación literaria considerando al escritor como un pensador creativo y diseñador de texto. Destaca que la creatividad no es un proceso aislado y propone una serie de ideas para explicar los pasos que seguimos los seres humanos al momento de escribir.

En su tesis doctoral Rafael Pérez y Pérez desarrolla un modelo en computadora, Mexica, que escribe pequeños cuentos sobre los Mexicas basado en las ideas sugeridas por Sharples.

2.4.6.1 Cómo escribimos

Para Sharples, la creación literaria es conceptualizada como un ciclo en el que alternan el ensimismamiento cognitivo y la reflexión. Un episodio creativo comienza con un conjunto de restricciones internas (el tema y retóricas: cómo presentarlo) y externas (el ambiente del escritor) que proveen un contexto para la creatividad lingüística.

Un escritor busca en su memoria ideas que satisfagan las restricciones y las escribe en prosa. El contexto dispara nuevas búsquedas en memoria para continuar el proceso. Este ensimismamiento en el texto que emerge puede ser interrumpido cuando no se encuentran ideas en la memoria, cuando el escritor percibe que el texto se aparta de las restricciones o cuando decide explorar un espacio conceptual para crear nuevo contenido. En dicho punto el escritor se detiene para analizar conscientemente el estado actual de lo escrito, ya sea revisando el material producido, revisando otros escritos, formando y transformando ideas y especificando qué nuevo material crear y cómo organizarlo. El proceso del ensimismamiento mental y la reflexión forman un ciclo productivo. Mexica modela el ciclo de ensimismamiento y reflexión demostrando

que el modelo es adecuado para especificar de forma muy general la producción de cuentos breves. Como Mexica no es un ser, sino un sistema, se limita a simular los procesos y restricciones mentales sin tomar en cuenta las restricciones externas.

Mexica no es un modelo de un escritor humano, sino una herramienta para explorar un conjunto de observaciones sobre la creación literaria.

2.4.6.2 Aspectos generales de Mexica

Mexica está dividido en dos partes principales: el Estado de Ensimismamiento y el Estado Reflexivo.

Durante el Estado de Ensimismamiento Mexica genera material guiado por las restricciones de contenido y retóricas evitando el uso de metas específicas o de información de la estructura del cuento.

Esta característica contrasta con sistemas anteriores que emplean metas explícitas o información de la estructura del cuento para generar sus salidas (e.g. Minstrel). Durante el Estado Reflexivo Mexica sale de estancamientos, modifica el cuento para que sea coherente y evalúa la novedad y lo interesante de la misma. Como un resultado de esta evaluación Mexica puede modificar las restricciones que guían la producción de material durante el Ensimismamiento. Así, los cuentos producidos por Mexica son el resultado de la interacción entre el ensimismamiento y la reflexión.

El sistema emplea una base de datos para crear en la memoria un grupo de estructuras que representan el conocimiento (de contenido y retórico) necesario para construir cuentos. La base de datos es llamada Historias Previas y está conformada por un conjunto de cuentos. En Mexica los cuentos consisten de 2 clases diferentes de componentes. La primer clase es formada por los elementos explícitos descritos en la narrativa, esto es, las acciones que pueden ocurrir en la historia del cuento tales como la princesa fue al bosque. La segunda clase esta formada por elementos tácitos (nunca son mostrados en el cuento) que representan relaciones emocionales entre los personajes y la tensión dramática producida en el cuento. Estos componentes tácitos son almacenados en una estructura llamada Contexto del mundo en el cuento. Esta estructura es usada como muestra para buscar en la memoria, durante el Estado de Ensimismamiento, las acciones que podrían ser la siguiente Acción en el cuento en progreso. Las estructuras Contexto pueden ser modificadas durante una búsqueda en la memoria a fin de obtener acciones interesantes y novedosas.

Para evaluar la novedad del cuento que se está generando el sistema la compara con las historias previas en términos de diferencias en el contenido. Para evaluar qué tan interesante es, Mexica compara el patrón de cambio en la tensión dramática del cuento que se está generando con los patrones de las historias previas. Mexica incluye un grupo de parámetros que pueden ser ajustados para controlar y modificar la estrategia de escritura del sistema de manera que el usuario puede experimentar con los diferentes aspectos del modelo.

2.5 IA En los Videojuegos

Por la complejidad inherente de los ambientes dinámicos es muy difícil lidiar con ellos, es por eso, que para su investigación se requiere de abstracciones o modelaciones de los mismos para poderlos manejar y comprender. El resultado de la creación de algunos de estos modelos relacionados con ambientes reales y dinámicos ha posibilitado la creación de juegos de computadora que simulan ambientes reales.

Estos se han implementado con variados objetivos que van desde el mero entretenimiento hasta el entrenamiento de personal especializado. En lo siguiente se hace una muy breve revisión de la IA utilizada en videojuegos.

2.5.1 Dungeon Keeper



Fig. 2.5 Pantalla del Videojuego Dungeon Keeper

Este es un juego que se acerca mucho al de estrategia en tiempo real como al de rol. El concepto es interesante, se trata de internarse en mazmorras que sirven de cobijo para monstruos, se pueden fabricar trampas, se pelea contra héroes y se gana oro con cada pelea. El papel del usuario es como el jefe de las mazmorras, es decir, es

él quien dirige la construcción de su guarida subterránea, desarrollando hechizos y usando las armas disponibles. Por otro lado los héroes intentan sabotear los planes de expansión o simplemente de enriquecimiento del usuario.

La inteligencia Artificial de este juego es una implementación de un proceso llamado “Clonación de comportamiento” el cual aprende del comportamiento del jugador. Los cerebros de los monstruos vienen ya con una experiencia de horas de juego de sus diseñadores y cada vez que un usuario intenta un nuevo ataque o truco el sistema lo registra y pasa a ser parte de su base de conocimiento. Es tan eficiente esta implementación que jugando en red pueden dejarse jugando solas a la criaturas aprovechando su comportamiento ya establecido.

Puede decirse que este juego es el más sofisticado en cuanto a IA se refiere, pues cada monstruo cuenta con 1500 bytes dedicados a su IA inicial con las capacidades de ver, oler y escuchar. Además de sentir miedo y huir. Las habilidades que poseen algunos de los monstruos las usan para dirigir a aquellos que no las poseen.

Por sus características este juego se le considera del tipo “God games” con una amplia variedad de personalidades y el grado de autonomía de sus agentes.

2.5.2 Field of Battle



Fig. 2.6 Pantalla del Videojuego Field of Battle

Es un juego de estrategia de guerra inspirado en los eventos tácticos que caracterizaron a la Primera Guerra mundial. Los jugadores intentan tomar el poder de Europa, Rusia y el Medio Oriente. La región de juego está dividida en secciones y la

implementación permite la producción y sabotaje de recursos lo que le genera un ambiente muy interesante.

La implementación de su inteligencia artificial esta basada en las Redes Neuronales. Lo que hace de este juego el primero en usarlas para tal propósito.

Su análisis de ambiente lo hacen con Redes Neuronales estocásticas de Hopfield. En el juego comercial no existe ningún archivo de inicialización de la Red Neuronal, de modo que esta está implantada en el interior de su código. Además el entrenamiento de la red no se hace en tiempo de juego sino al comenzar este. En esencia, la computadora de acuerdo a la situación actual se adapta y decide que unidad (agente) va a usar y donde la va a colocar, la red entonces evalúa todos los posibles lugares en donde se puede colocar esta unidad y lo hace aunque en ocasiones hace la decisión de forma aleatoria.

Un gran problema es que el sistema se auto ajusta a la situación actual sin una evaluación del plan original. Lo que se ha pensado hacer para solucionar el problema es correr –antes que la red entre en acción– un algoritmo genérico de estrategia que actualice los datos necesarios, haga una simplificación de la situación y entonces mediante un proceso de decisión seleccionar entre actuar o no hacerlo de acuerdo a la situación.

2.5.3 SWAT



Fig. 2.7 Swat, Un videojuego de estrategia.

Este es un juego de estrategia en tiempo real del género de acción. Este usa interesantes técnicas de IA para el control de los criminales que toman lugar en el juego.

En general el juego esta basado en eventos actuales, se desarrolla en un ambiente tal que la última versión de este juego simula acontecimientos del año 2006. La ciudad de Los Ángeles se encuentra tomada por el caos y amenazada por crecientes bandas callejeras. El jugador toma entonces el papel de oficial de un grupo de élite SWAT y su misión es restaurar el orden.

SWAT usa Inteligencia Artificial de nivel elevado que permite a los agentes participantes a tener comportamientos dependiendo de sus personalidades y habilidades aunado a su percepción del ambiente, de modo que los diferentes personajes no reaccionan de la misma forma ante un mismo estímulo y condiciones idénticas, sino que cada personaje puede reaccionar de diferente forma cada vez que se encuentre en las mismas condiciones.

Si Inteligencia Artificial esta dividida en dos módulos bien diferenciados: 1)IA Estratégica y 2) IA Táctica.

2.5.3.1 IA Estratégica

Es la que toma el control del oponente del jugador. Cuando el jugador selecciona controlar el grupo SWAT, la IA se encarga de la coordinación del grupo, es decir hace que trabajen juntos anticipándose a movimientos lógicamente inmediatos simulando un trabajo en equipo. De igual forma las acciones de los terroristas y su coordinación depende de sus metas, aspectos que controla este módulo inteligente.

Idealmente esta diseñado para hacer creer al usuario que se enfrenta a un oponente humano que reacciona dinámicamente. Por ejemplo, en el caso de que el sistema controle a los terroristas, puede enviar a uno del grupo a buscar una ruta de escape, o en el caso de controlar a el grupo SWAT entonces anticiparse a acciones como la anterior planeando emboscadas.

2.5.3.2 IA Táctica

Esta controla la personalidad y comportamiento particular de cada uno de los personajes, les dice cómo, cuando y donde optar por acciones como explorar la zona, tomar rehenes o escapar corriendo. La IA Táctica toma el control completo a nivel “atómico”, esto es, desde acciones como “Muévete un paso hacia el Norte” o “Envía un mensaje de radio” hasta acciones de niveles superiores como “Morir” o “Recoge una tarjeta de teléfono” y “Muévete al punto x, y”. En el nivel mas alto se encuentra acciones como “Rastrea la zona” o “Cuida los accesos a esta zona”.

En el nivel más alto de la IA táctica se usa Lógica difusa. Por ejemplo en la siguiente situación: La unidad SWAP se encuentra en cierta posición y empieza a ser bombardeada. Esta unidad puede tener varios comportamientos como: Huir, Solamente retroceder, o responder al ataque, entre otras. Cada uno de los miembros de la unidad tienen una personalidad predefinida que consiste en cuatro características mentales: Agresión, Valentía, Inteligencia y Cooperación. Entonces cada integrante de la unidad responde de acuerdo a su personalidad pero ligeramente influenciado por aleatoriedad para simular realismo. El motor Lógico difuso toma en cuenta ambos aspectos para tomar una decisión de la siguiente forma: Se toma en cuenta un evento “disparado” por el ambiente del cual forman parte los agentes y reciben este como entrada, este evento entra entonces a una serie de reglas difusas del tipo: “Si la Agresión es alta y la Valentía es alta entonces Responde al ataque”.

La clave para el uso de la Lógica difusa es el hecho de que los miembros de una unidad pueden parcialmente pertenecer a varias unidades y las unidades no son iguales, por ejemplo, suponiendo que algún miembro de una unidad tenga una Agresión = 5 la cual puede considerarse alta, pero que en otro grupo en la cual la agresividad de todos los miembros es sumamente alta, entonces la Agresividad = 5 puede ser baja o media. Entonces para tomar una decisión el motor difuso evalúa la personalidad de cada uno de los integrantes de una unidad en términos de baja, media o alta con relación a la unidad a la que pertenece en ese momento el agente sobre el cual se tiene que decidir.

Resumen

Con el objetivo de ofrecer un panorama de los avances de la Inteligencia Artificial relacionados a este trabajo, en este capítulo se revisaron algunos trabajos e investigaciones así como ejemplos específicos sobre la Computación Creativa, el proceso creativo como tal y las técnicas usadas actualmente en la generación de Videojuegos.

Además se hizo una breve reseña de Mexica, que es un ejemplo de la implementación del modelo Ensimismamiento-Reflexión (E-R), el cual es la base de este trabajo.

Capítulo 3

Adaptación del modelo

Ensimismamiento-Reflexión a la generación de estrategias en un videojuego de computadora

En general, lo que trata este capítulo es la explicación detallada del trabajo de implementación de la adaptación del modelo Ensimismamiento Reflexión al videojuego de estrategia que fue desarrollado como parte de este proyecto de tesis. Así mismo se presenta la investigación hecha para realizar esta adaptación, los resultados de esta y su implementación.

En general se da una visión amplia del juego implementado, sus reglas, sus modos de juego, y una descripción de su interfaz. Así mismo se presentan todos los programas auxiliares desarrollados para el procesamiento y análisis de la información obtenida en una fase de experimentación, su propósito así como su funcionamiento. Así mismo se le da seguimiento a la lógica seguida para poder establecer y definir los elementos necesarios para poder hacer trabajar la teoría del modelo del proceso creativo usado en este proyecto.

3.1 Especificación de las Reglas del Juego

Hay dos formas únicas de ganar el juego

1. Agotando la resistencia de la base enemiga.
2. Agotando la resistencia de todos los integrantes del equipo enemigo.

Ambas formas sugieren la constante adquisición de materiales y su uso estratégico para alcanzar cualquiera de los dos anteriores objetivos.

Para ganar el juego están disponibles dos tipos de recursos que son agentes (como robots y guardianes) y “materiales” susceptibles de ser almacenados en la base de control. A estos últimos recursos se les ha llamado **recursos abstractos** por la razón de no existir como una entidad (como un agente propiamente dicho) dentro del juego, sino que es usada una representación numérica para indicar su existencia y dar por enterado de ello a los agentes quienes hacen uso de ellos.

Dentro de los recursos abstractos están el material (madera, piedra o metal) almacenados que se usan para la “compra” de otros recursos disponibles. También están las tarjetas de poder dentro de este rublo. Existen sin embargo recursos que son representados como abstractos, pero el momento de usarlos se convierten en recursos de tipo agente. Tal es el caso de las Bombas y Antibombas. Estas, una vez adquiridas (compradas usando material almacenado) son un recurso abstracto, lo cual indica que los agentes saben que “existen” almacenados en la base de control. Cuando se decide hacer uso de uno de ellos, entonces se convierten en recursos de tipo agente pues se hace evidente su existencia en el ambiente de juego y se da a conocer, por lo que adquieren la capacidad de comunicarse con los agentes con quienes puedan intercambiar información.

Debe decirse que los recursos abstractos se usan como representación de lo que sucede en la mente humana cuando se tiene conciencia de la existencia o disponibilidad de algo, de manera que, para efectos de este trabajo, se ha representado esta existencia con cantidades o valores numéricos los cuales deben ser administrados también por el desarrollo lógico del juego. Esto quiere decir, que si se almacena constantemente algo en la base de control este tiende a aumentar, por lo que el valor numérico que representa su existencia también aumenta y viceversa.

3.2 La Interacción Jugador-Ambiente de Juego

La interacción de un jugador con el ambiente de juego es mediante una interfaz que permite controlar uno de los dos equipos contrincantes. Cada uno de los equipos está constituido por tres agentes capaces de recibir ordenes por parte de quien los controla a través de una interfaz gráfica que puede ser manejada mediante el teclado de la computadora.

Además de estos tres agentes, los cuales inicialmente componen el equipo, como se explicó en la sección anterior, es posible que el jugador adquiera otros agentes y se agreguen a su equipo. Estos últimos también pueden recibir órdenes de la misma forma que los que inicialmente componen el equipo.

La interacción con el ambiente por parte de un jugador también depende de la clase de acciones que se quiera realizar puesto que las decisiones tomadas por el jugador y la posibilidad de realizarlas están sujetas a la naturaleza de los agentes que forman parte del juego y que representan el medio por el cual el usuario actúa con este.

Como se mencionó en la sección anterior, cada uno de los agentes tienen habilidades y capacidades diferentes. Por lo anterior se han clasificado todos los agentes del juego en dos categorías:

1. Agentes reactivos.
2. Agentes racionales.

Los **agentes reactivos** son aquellos agentes que solo tienen la capacidad de recibir mensajes del ambiente y emitir o no un mensaje como respuesta, sin ninguna clase de procesamiento de la información recibida. Por otro lado los **agentes racionales** son aquellos que además de tener la capacidad de recibir y emitir mensajes pueden hacer una evaluación o procesamiento de la información que están recibiendo.

Dicho esto, debe especificarse que los únicos agentes racionales en el juego son los agentes de tipo soldado, es decir, los integrantes de cada uno de los equipos.

De esto se desprende entonces una clasificación de las acciones que pueden realizar los agentes racionales.

1. Acciones cognitivas (tipo agente)
2. Acciones de interacción con el ambiente.

Las **acciones cognitivas**, como su nombre lo indica, simula el procesamiento de información recibida que siempre tienen lugar en un cerebro humano, que como bien es sabido, todo el tiempo estamos recibiendo información por parte del ambiente, por lo que tiene que ser filtrada y procesada para poder reconocer la situación actual y tomar decisiones en consecuencia. En este trabajo, las acciones cognitivas es un proceso de evaluación de la situación en la que se encuentra un agente con respecto a su ambiente, y estas tienen lugar siempre que presente un evento.

Específicamente en este trabajo, se entiende por “evento” a un cambio en el estado particular de un agente o en el estado general del ambiente, por ejemplo, la reducción en una o varias unidades de la resistencia de un soldado o la aparición de un tornado o una bomba, etc.

Aquí tiene que mencionarse que existe otro tipo de acciones cognitivas que son de tipo “modelo cognitivo”. A diferencia de las acciones cognitivas “tipo agente” las tipo modelo cognitivo tienen por finalidad “decidir” a quien se le asignará una acción una vez que la implementación del modelo Ensimismamiento Reflexión a “tomado una decisión”. Estas últimas se explicarán en más adelante.

Por otra parte, las **acciones de interacción con el ambiente** son en realidad la representación gráfica y visual de las acciones que son tomadas por quien controla a uno de los equipos en el juego. Estas acciones a su vez se dividen en dos categorías:

1. Acciones de tipo agente.
2. Acciones de tipo usuario.

Las **acciones de tipo agente** son aquellas que son manifiestas gráficamente en pantalla con un movimiento del agente que ha recibido una orden. Entre esta clase de acciones está el Ir por material, atacar, bombardear, etc. Solo por dar un ejemplo, la acción: “Ir por madera” es manifiesta mediante un desplazamiento visual del agente que ha recibido la orden del centro de abastecimiento seleccionado a la base de control, lo que resulta en una comunicación entre el agente que recibió la orden y los agentes involucrados en la acción (en este caso, el centro de abasto y la base de control), pues tiene que hacer “peticiones” de material al punto de abasto y tiene que hacer “entregas” en la base para su almacenamiento.

Las **acciones de tipo usuario** representan las transacciones para adquirir recursos. Específicamente es la “compra” de recursos usando como medio de

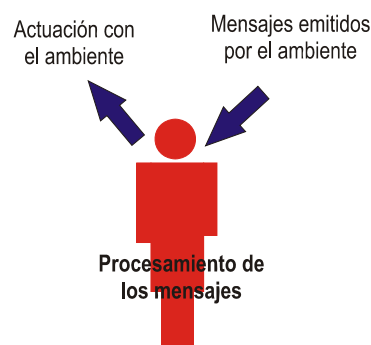
intercambio el material almacenado en la base de control. Para estas acciones se requiere tener “conciencia” de los recursos disponibles, pues si el valor de un recurso que se desea adquirir es mayor a la cantidad de material disponible o almacenado en la base, lógicamente no se podrá adquirir. Este tipo de acciones son manifiestas con el aumento de los valores numéricos que representan los recursos almacenados dentro de la base.

3.3 Los Agentes Racionales

La especificación del juego y el ambiente mismo demanda una constante comunicación entre todo o existente dentro de él. Así mismo, puesto que la implementación de la Inteligencia Artificial requiere saber que es lo que existe en cada momento y manejar a cada una de las entidades existentes como un individuo totalmente independiente, se tuvo que buscar la forma mas apropiada para implementar dichas características necesarias para el desarrollo del juego. Todo esto desembocó en el seguimiento de la teoría de los Agentes Inteligentes.

De acuerdo con Stuart Russell [8] Un agente es cualquier cosa capaz de percibir su medioambiente con la ayuda de un sistema de percepción y actuar en ese medio utilizando actuadores.

Siguiendo este concepto se implementaron diferentes agentes, pero en general, la especificación de estos se puede ilustrar con la siguiente figura:



Como se puede notar, un agente tiene la capacidad de comunicarse con su ambiente en dos sentidos, percibiendo los eventos que suceden en el ambiente y actuando con cada uno de las demás entidades existentes a su alrededor.

De acuerdo a la naturaleza del agente y su rol dentro del juego se determina la clase de señales que puede percibir y el tipo de actuación que puede tener con el ambiente. Debe decirse que los Agentes con mas capacidades de interacción son los agentes de tipo Soldado que integran cada uno de los equipos pues están diseñados para tener “conciencia” de todo lo que existe en un momento determinado en el ambiente.

La comunicación entre todos los agentes y su medioambiente se ha implementado usando el recurso de envío de mensajes que proporciona el sistema operativo Windows que a grosso modo se tratan de mensajes identificados por un numero entero que es especificado por el programador. Los mensajes son emitidos dado un evento. Una vez emitido un mensaje, uno o todos los agentes existentes en ese momento en el ambiente es/son capaces de reconocer el mensaje y entonces reaccionar ante este. Por ejemplo:

Dado el evento “Aparición de Tornado” todos los agentes soldados son enterados de este evento y empiezan con una actividad de evaluación para “saber” en que situación se encuentran con respecto a esta nueva situación en el ambiente.

3.4 El Manejo de Mensajes Emitidos en el Ambiente de Juego

Puesto que son muchos los mensajes que se manejan dentro del ambiente (tantos como 125) se hizo necesario desarrollar un software para administrar cada uno de estos mensajes. Este software tiene por propósito almacenar el número identificador de cada mensaje, el sentido de la comunicación (quien emite y quien recibe) y su interpretación. La interfaz de este programa se ve como en la siguiente figura

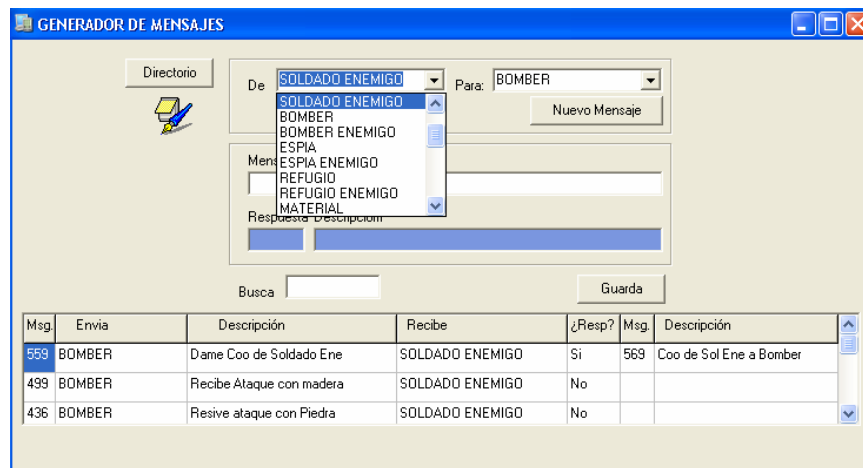


Fig. 3.1 Interfaz del programa “GeneraMensajes.exe”.

Siguiendo con esta implementación es que se pudo lograr hacer a cada uno de los tipos de agentes diferentes a los demás, con capacidades distintas y específicas y sin embargo capaces de comunicarse con el medio ambiente y los demás agentes. Por ejemplo, los agentes de tipo soldado son tres: El soldado, el Bombardero y el espía, en donde todos pueden saber acerca de la presencia de los otros dos, y pueden intercambiar información como el estado en el que se encuentran, si están atrapados o requieren ayuda y sin embargo tienen cada uno habilidades diferentes. Esto es, el Espía por ejemplo es el único agente que puede “robar” de la base enemiga, y el bombardero es el único capaz de manejar las bombas. Sin embargo, todos los agentes de tipo soldado comparten un comportamiento básico que les permite intercambiar información con los centros de abastecimiento, con las bases de control con los enemigos... etc. En general con todo el ambiente.

Y son todas las anteriores características de los agentes las que son usadas por la implementación del modelo creativo para la generación de una estrategia de juego.

3.5 El Tablero de Juego

El ambiente de juego está representado por un área grafica en donde se desarrolla una partida. Esta se puede ver en la siguiente ilustración:



Fig. 3.2 Interfaz del videojuego Guns.

Esta interfaz gráfica se divide en dos partes esenciales:

1. El área de juego, en donde se mueven los agentes y tienen lugar su interacción.
2. El área de control.

En el **área de juego** es donde se desarrollan las estrategias por parte de los jugadores. Es donde se ven los movimientos de los agentes y donde tiene lugar la interacción de estos con todo el ambiente. Inicialmente, cuando se comienza una nueva partida existen: 2 puntos de abastecimiento de madera, 2 puntos de abastecimiento de piedra y dos puntos de abastecimiento de metal, un equipo rojo compuesto por tres soldados (1 Soldado, 1 Bombardero y 1 Espía), y un equipo azul, cada uno con una trampa a su disposición y una base de control para cada uno de los equipos.

En el centro del área de juego hay especificada un área de peligro la cual, al ser “pisada” por alguno de los agentes aparece un agente “Tornado” el cual representa una amenaza para cualquiera de los soldados. Es en esta misma área donde aleatoriamente (en tiempo y espacio) aparecen las tarjetas de poder, las cuales son adquiridas por los equipos bajo su propio riesgo.

En el **área de control** se encuentra la interfaz con el usuario que muestra el estado general del equipo. Esto es, donde se muestra la resistencia actual de los agentes soldado, de la base de control y de los agentes adquiridos o comprados. Así mismo se muestra la cantidad de recursos almacenados de cada equipo.

En esta misma área se encuentran algunos controles que permiten especificar variables del juego tales como el modo de juego (novato, medio, experto), y permite ver los cronómetros que indican la duración de un evento como el encendido de una bomba hasta su explosión, la aparición de una tarjeta o el congelamiento de uno de los equipos.

Así mismo se pueden ver controles que fueron puestos para el estudio del ambiente y que fue necesario para poder desarrollar la implementación del modelo a este videojuego. Tales controles permiten mostrar interfaces que contienen datos como los contextos de un agente particular en un momento dado, hacer una pausa total en el juego, cargar un contexto almacenado en un archivo de texto o crear uno en tiempo de juego. También se incluyen controles que permiten la configuración del teclado para la manipulación de los agentes, hacer un paso a paso y activar el modo de juego “usuario contra IA”.

Es en esta misma área en donde se implementó la interfaz mediante el cual un usuario puede asignarle ordenes a cada uno de los agentes y hacer sus compras de recursos y tomar algunas decisiones como colocar las trampas en un punto determinado.

3.6 El Esquema General de la Implementación.

El esquema general de la implementación y adaptación del modelo puede explicarse mediante la siguiente ilustración

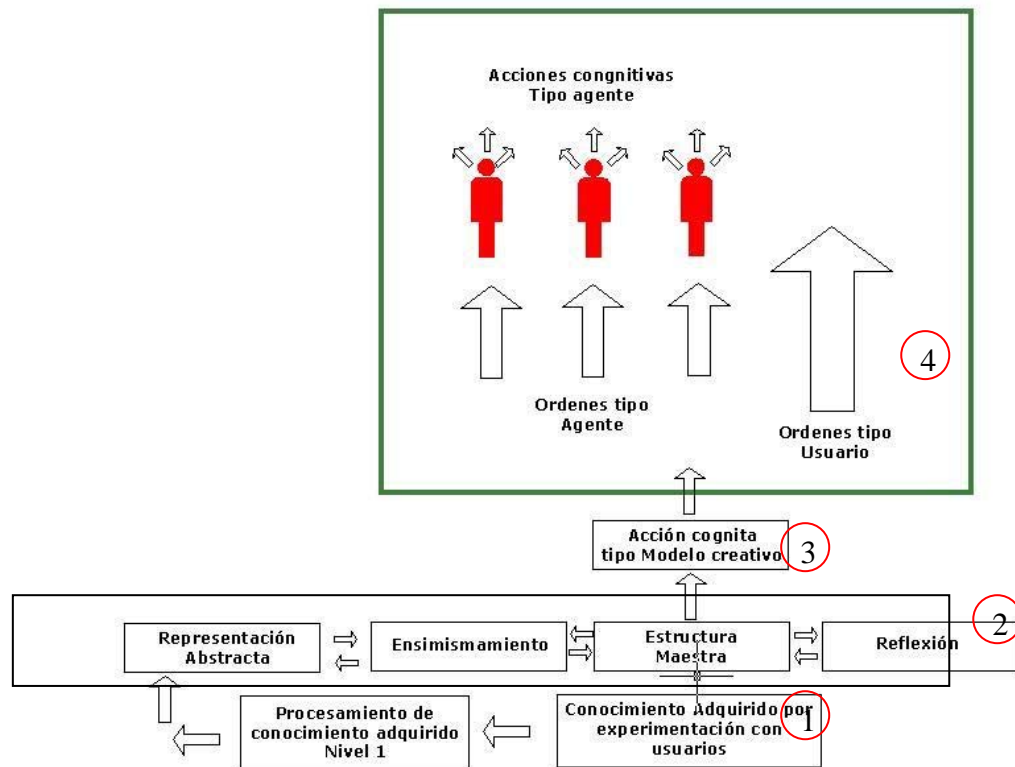


Fig. 3.3 Esquema general de la adaptación del modelo Ensimismamiento-Reflexión al videojuego de estrategia.

Básicamente la implementación del modelo consta de cuatro partes bien diferenciadas:

1. Adquisición y procesamiento de la base de conocimiento
2. La implementación del modelo Ensimismamiento-Reflexión
3. Implementación de la interacción con el ambiente de juego.
4. El ambiente de juego.

3.7 Descripción Detallada de la Adaptación del Modelo

El modelo Ensimismamiento-Reflexión, en general, establece que un proceso creativo puede simularse mediante la interacción cíclica de dos estados particulares: 1) Ensimismamiento y 2) Reflexión.

En el primero, se hace uso de experiencia previa para obtener material que “puede” ser útil en la generación del resultado final buscado, que en este caso, es una estrategia de juego que pueda guiar hasta la posible victoria al equipo controlado por la Implementación de este modelo, a lo que se le llama la parte Inteligente. Por otra parte el segundo estado se encarga de validar el material generado en el estado anterior.

Ambos estados trabajan en ciclos pasando de uno a otro para generar la estrategia de juego, haciendo uso de transformaciones de estructuras que usan la información disponible y necesaria en tiempo de juego junto con aquella que se tiene como experiencia previa y finalmente pasando el control a procedimientos que asignan las ordenes a cada uno de los agentes existentes en el juego, y ayudándose por acciones de evaluación que como se explicará en este documento se les ha llamado “Acciones cognitivas”.

El proceso completo para la generación de estrategias en la adaptación del modelo usado en este trabajo consta de tres etapas que se pueden describir como sigue:

1. Obtención de experiencia. (Experimentos hechos con usuarios de donde se recoge y registra todo un juego (en archivos de texto) con algunos comentarios al respecto de los criterios para tomar decisiones en tiempo de juego)
2. Procesamiento de la experiencia Previa (Creación de la Representación abstracta). Esto es, la codificación de la experiencia obtenida en el inciso anterior en términos de tensiones, relaciones y causas usando una terminología particular para su uso durante la intervención del modelo Ensimismamiento-Reflexión.
3. Intervención del modelo del proceso creativo. Por la dinámica del juego, se requieren de acciones de evaluación a semejanza de lo que hacen los usuarios durante una partida con el juego. Esta evaluación es un proceso continuo y que influye directamente sobre la toma de decisiones, y que finalmente se valen de los ciclos entre los dos estados característicos del modelo usado para generar una estrategia de juego y un plan completo para jugar y tratar de ganar la partida. Debe decirse que ambos, tanto la estrategia como el plan de juego cambian de acuerdo a las circunstancias, pues el juego es dinámico.

En este contexto, y por la implementación particular de este proyecto, se entiende por estrategia al modo de juego o criterios predominantes que se adoptan para tratar de ganar una partida. El plan de juego son el conjunto ordenado de acciones tales que ayudan a alcanzar y ejecutar la estrategia de juego.

Más específicamente. Una estrategia, en la implementación de este trabajo, esta formada por tres componentes:

1. Estrategia. (Especifica los principales criterios, o modo de juego).

2. Objetivo final (Especifica como se pretende ganar el juego)
3. Actitud (La actitud con la que se va a atender las anteriores especificaciones)

A su vez la estrategia tiene cuatro componentes

E.1. Modo: ataque a Base enemiga (Continuos/Oportunos/Nunca).

E.2. Modo: ataque a soldados (Continuos/Oportunos/Nunca).

E.3. Modo: atención a tarjetas de poder (Continuos/Oportunos/Nunca).

E.4 Modo: Uso de equipo. (Total/Parcial).

Para entender todos estos parámetros se hacen los siguientes comentarios:

Se puede suponer que un Jugador quiere terminar el juego con ataques directos (mediante bombardeos) sobre la base enemiga para acabar con la resistencia de esa base y ganar el juego. Tal vez no quisiera hacer uso de las tarjetas de poder para congelar al enemigo, pero que si la situación es muy favorable como para obtener algunas, puede darles atención. Pero en general, tal vez quisiera terminar rápido el juego, en otras palabras, ser totalmente ofensivo.

Para este modo de juego se puede tener entonces una estrategia de la siguiente forma:

Estrategia	
Ataque Base enemiga	Continuos
Ataque soldados enemigos	Nunca
Atención tarjetas de poder	Oportunos
Uso del equipo	Total
Objetivo final	
Objetivo Final	Bombardera Base
Actitud	
Actitud	Ofensiva

Fig.3.4 Tabla de resumen de un ejemplo de la especificación de los parámetros que definen una estrategia de juego en el videojuego Guns.

3.7.1 La Estructura Maestra

Para el manejo de esta in formación se ha creado una estructura general la cual es llamada en la implementación de este proyecto como “Estructura Maestra” y sobre la cual el modelo principalmente trabaja. Esquemáticamente puede verse como sigue:

Estrategia	A. Base + A. Soldado + A. Tarjetas + U. Equipo		
Objetivo Final	Obj. Final		
Obj. Particulares (Completo = false)			
Objetivos	Estado	Acciones Posibles	Numero Acciones
Obj. P 1	Cumplido/Sin cumplir 1	[Set acciones posibles p/ Obj P. 1]	Set 1 No
Obj. P. 2	Cumplido/Sin cumplir 2	c/acción con su tensión tota/sol, bom, spy/1 y	
...	...	CRUZA/sol, bom, spy/	Set 2 No
Obj. P. n	Cumplido/Sin cumplir n	[Set acciones posibles p/ Obj P. 2]	...
		...	Set n No.
		[Set acciones posibles p/ Obj P. n]	
Plan (Completo = false)			
Acciones		Tipo	
Acción Seleccionada 1			
Acción Seleccionada 1			
...			
Acción Seleccionada n			
Actividades Equipo (Completo = false)			
Soldado	Bombardero	Espía	
Acción Anterior	Acción Anterior	Acción Anterior	
Acción (Soldado)	Acción (Bombardero)	Acción (Espía)	

Conjunto de acciones Posibles			
Acción 1: Acción 1. (Tensión Base)			
Tensión (SR)		Cruza	
Tensión Soldado E	Intero	Zona	Cruza Sol E
T. Bomber E	Intero	Tomado	Cruza Sol E
T Espia E	Intero	Trampa	Cruza Sol E
...		Bomba	Cruza Sol E
		Guardián	Cruza Sol E
		Cruza Bombardero E	
		Zona	Cruza Sol E
		Tomado	Cruza Sol E
		Trampa	Cruza Sol E
		Bomba	Cruza Sol E
		Guardián	Cruza Sol E
		Cruza Espia	
		Zona	Cruza Sol E
		Tomado	Cruza Sol E
		Trampa	Cruza Sol E
		Bomba	Cruza Sol E
		Guardián	Cruza Sol E

Fig. 3.5. Composición interna de la Estructura Maestra.

El fragmento de código que representa esta estructura es el siguiente:

```
// esta es la estructura que representa a el plan maestro
EsqueletoMaestro = record
    Estrategia      : contieneEstrategia;
    ObjFinal        : string;
    Actitud         : string;
    ObjParticulares : array of objetivoParticular;
    ObjPartiCompleto: boolean;
    Plan            : array of accionPlan;
    AccionesEqu     : ActividadesEquipo;
    AccionesEquCompleto : boolean;
end;
```

Fig. 3.6 Representación de la Estructura Maestra en Código

Esta es la estructura principal y general sobre la cual trabaja la interacción del modelo Ensimismamiento-Reflexión. La intervención de éste es debida a banderas activadas y desactivadas por los eventos en el juego.

La estructura Maestra es totalmente dinámica (se redimensiona en tiempo de juego) de acuerdo a las necesidades del modelo y a las acciones cognitivas.

En seguida se explica cada una de las partes que conforman a la Estructura Maestra.

Las 5 partes esenciales que conforman la Estructura Maestra son las siguientes:

1. Espacio de Estrategia.
2. Espacio de Objetivo Final.
3. Espacio de Objetivos Particulares.
4. Espacio de Plan.
5. Espacio de asignación de acciones.

3.7.1.1 La Sección de Estrategia

En este espacio se almacenan los cuatro elementos que conforman la estrategia y que se mencionaron anteriormente, a saber, el modo de juego con respecto a 1) Ataques a la base enemiga, 2) Ataque a agentes enemigos, 3) Atención a la parición de las tarjetas de poder y 4) El uso del equipo (Total o parcial) para cada acción o actividad.

El parámetro que define el modo de juego respecto a los Ataques a la base enemiga, es uno de tipo cadena de caracteres, el cual puede tomar el valor de Continuos/Oportunos/Nunca. Si el valor de este componente de la estrategia toma el valor de “Continuos” eso significa que su principal interés, dadas las condiciones necesarias (teniendo el material necesario almacenado para ejecutar un ataque a la base enemigo) no se dudará en ejecutar ese ataque.

Por otra parte, un valor “Oportuno” implica que las condiciones actuales con respecto a un equipo en particular son favorables (de riesgo mínimo, o de condiciones óptimas) para realizar el ataque. Este valor de este parámetro da pie a una evaluación de la situación actual predominante en el juego mediante una de las acciones cognitivas (en este caso particular, la acción “EvaluaSistema” implantada de manera particular en cada uno de los agentes de cada equipo) que tiene por propósito obtener los valores particulares de un agente que representan tensiones (valores de peligro por proximidad de amenazas al agente en cuestión y por trayectoria, que es la evaluación del “posible”

camino a seguir para ejecutar una acción, para ver si en este se puede encontrar con una amenaza de peligro inminente, como una trampa).

Por último, un valor “Nunca” de este parámetro indica que JAMÁS se hará el intento por realizar acciones que tengan que ver con ataques a la base enemiga.

El criterio que define estos tres posibles valores son aplicables a cada uno de los tres primeros elementos que definen el Espacio de Estrategia en la estructura Maestra, a excepción del cuarto.

Este último parámetro (Uso de equipo) puede adoptar dos posibles valores: Parcial/Total. Un valor de “Total” identifica un modo de juego en donde todo el equipo va a estar concentrado en una sola tarea, es decir, por ejemplo, si la acción a seguir tiene por propósito almacenar un materia específico en la base de control, entonces TODOS los agentes existentes hasta este instante estarán concentrados en ejecutar acciones que cumplan con ese propósito. Por otra parte, un valor “Parcial” indica que el equipo estará dividido atendiendo acciones con diferentes propósitos

3.7.1.2 La Sección de Objetivo Final

En este espacio de la Estructura Maestra se almacena un parámetro que define “La forma” en que el equipo “quiere” ganar la partida. Esto es, hay dos formas distintas de ganar el juego, ya sea llevar a cero la resistencia de la base enemiga o llevando a ceros la resistencia de todos los agentes enemigos. La decisión de cómo se pretende ganar el juego, con cualquiera de estas dos formas, es representada por este parámetro que es almacenado en el Espacio de objetivo Final.

3.7.1.3 La Sección de Objetivos Particulares.

Para entender el contenido de este espacio en la Estructura Maestra, se debe explicar un poco de la dinámica del juego.

Mediante la implementación del Modelo Ensimismamiento Reflexión para la generación de la estrategia de juego es que se obtiene material (conjunto de acciones) tal que pueda llevar a la posible victoria del equipo controlado por la IA. Cada una de estas acciones está predefinida por las reglas y el ambiente de juego (lo que es o no posible hacer dentro del ambiente de juego).

La mayoría de estas acciones tienen precondiciones para poderse ejecutar, así como poscondiciones.

Por lo tanto, teniendo ya generadas las acciones para “actuar” conforme a la estrategia definida, habrá que asegurarse de que estas acciones son posibles de realizar. Para tener certeza de ello, tienen que alcanzarse las precondiciones necesarias para poder ejecutar las acciones. A estas precondiciones necesarias se les almacena en este espacio y entonces llegar a ser los Objetivos particulares.

3.7.1.4 La Sección de Plan de Juego.

La aplicación del modelo a la dinámica del juego tiene como salida final la generación de las acciones adecuadas para poder ganar el juego apegándose al objetivo final. Estas acciones, son almacenadas en el espacio de la Estructura Maestra, llamado Plan.

3.7.1.5 La Sección de Asignación de Acciones

La asignación de cada una de las actividades contenidas en el Espacio de Plan de la estructura Maestra, debe ser asignada a los agentes para que estos a su vez las lleven a cabo apegándose a la estrategia ya explicada anteriormente, tratando de alcanzar el Objetivo final, apegándose al plan que es guiado indirectamente por los Objetivos particulares.

La asignación de las acciones se hace mediante una acción de tipo cognitiva, la cual evalúa los valores de todos los parámetros que definen la estrategia de juego, pero sobre todo por el parámetro de la estrategia llamado “Uso de equipo”.

Respetando todos estos parámetros e influenciado por las condiciones actuales del ambiente es como se asignan las tareas.

Debe decirse que el que una acción ocupe este espacio no quiere decir que los Agentes acaten la orden, mas bien, la asignación de acciones pone en su lugar la siguiente acción a realizar para cada agente presente (vivo) en el tablero de juego.

3.8 Etapas de la Implementación de la Adaptación del Modelo Ensimismamiento-Reflexión

En seguida se explican las tres etapas de la implementación de la adaptación del modelo Ensimismamiento-Reflexión en este proyecto.

3.8.1 Etapa 1: Obtención de experiencia Previa.

Este trabajo consta de una serie de pasos que inicia con la recolección de los datos durante una fase de experimentación y termina con la obtención de la Representación Abstracta de todo ese conocimiento adquirido.

Para realizar esto se requieren de varios módulos cuya interacción se puede explicar mediante la siguiente ilustración:

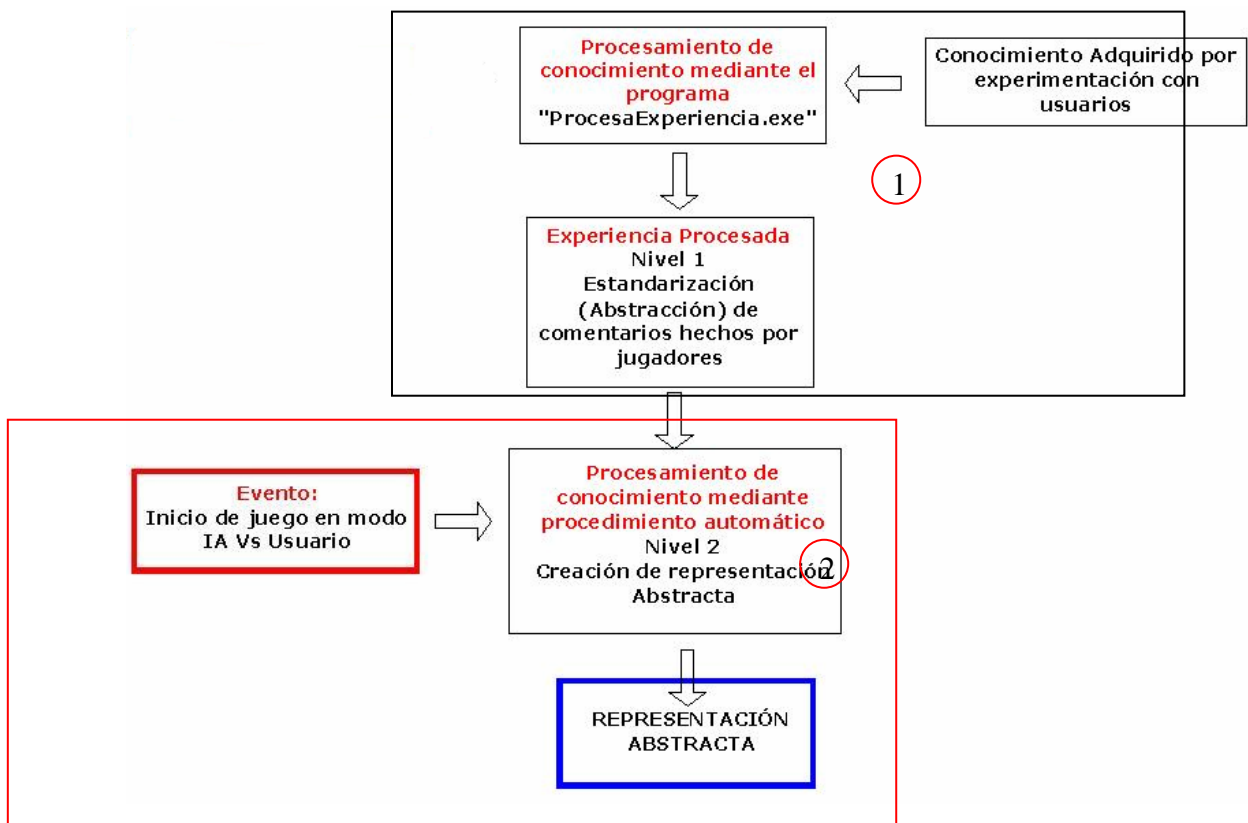


Fig. 3.7 Esquema del Procesamiento de la experiencia Previa

La obtención de la experiencia previa usable por la implementación del modelo puede dividirse en dos fases:

1. La adquisición y el preprocesamiento de la información compilada en la fase de experimentación.
2. La obtención de la Representación Abstracta.

El primer paso se hace mediante experimentación directa con usuarios y la información obtenida es guardada en un formato de texto.

La segunda fase inicia cuando estando en el modo de juego “IA Vs Usuario” se inicia un juego nuevo.

La primera fase se hizo en periodos de prueba con la ayuda de jugadores voluntarios, los cuales no tenían ninguna experiencia con el juego, sino que inicialmente se les dio una muy breve explicación de la dinámica de este.

Después de algunos experimentos se puede decir que llegaron a ser jugadores “expertos” del mismo, pues las reglas son sencillas y específicas.

La experimentación se hizo en ambas etapas, es decir, durante aquella en la que se les consideró novatos a los voluntarios y la etapa de expertos. Sin embargo, no se hace diferencia alguna en la selección de la experiencia obtenida, en general, toda partida guardada durante esta etapa se considera simplemente como eso, como experiencia previa, sin ningún criterio de preferencia de alguna sobre la otra.

La experiencia se registra activando las opciones pertinentes en el tablero de juego. Esto significa que durante el inicio del juego se especifica la opción de “Guardar partida”, de modo que cada vez que se toma una decisión o se ejecuta una acción, se guardan los datos completos de lo que sucede en el tablero, además de algunos comentarios por parte de los usuarios.

Un archivo de experiencia previa se ve como sigue:



```
listo - Bloc de notas
Archivo Edición Formato Ver Ayuda
Partida jugada el 03/08/2006 09:35:23 p.m.
=====
CONTEXTO_SOLDADO(Pos(40,35), Bomber(55,35): 15, Espia(70,35): 30, Sol_Ene(920,475): 984, Bom_Ene(935,475):
997, Esp_Ene(950,475): 1011, Tornado(No): , Carta(373,263): 404, DCarta(13,287,63,87), Guardian(No): , Guar_Ene
(No): , Robot(No): , Robot_Ene(No): , Trampa(10,10): 39, Trampa_Ene(1000,550): 1089, Bomba(No): , Refugio(No):
, Ref_Ene(No): , Mad_1(950,70): 891, Mad_2(100,500): 469, Met_1(460,280): 486, Met_2(520,240): 522, Pie_1(490,55):
450, Pie_2(510,415): 604, Base(20) Base_Ene(1000) Tsole(S1,1,984) Tbone(S1,1,997) Tespe(S1,1,1011) Ttor
(No,0,0) Tcarta(S1,1,404) Tbone(No,0,0) Tguare(No,0,0) Ttrae(S1,1,1089) Trefe(No,0,0) Tmad1(1,891) Tmad2
(1,469) Tmet1(1,486) Tmet2(1,522) Tpie1(1,450) Tpie2(1,604) TENSIONES(Vr: 1, Sr: 13)
xxxxxxxxxx
EDO_SOLDADO(resist: 5, Nadie me ataca, No ataco, Trampa: iLibre!, Guard: iLibre!, YENDO_POR_TARJETA, Desarmado)
BASE(Madera(0), Metal(0), Piedra(0), Bombas(0), Tarjetas(0), AntiBombas(0), Vr(4), Sr(43), NEC_MADERA(3), NEC_METAL
(3), NEC_PIEDRA(3), NEC_MATERIAL(9), COMP_MADERA(0), COMP_PIEDRA(0), COMP_METAL(0), COMP_ROBOT(0), COMP_BOMBAS
(0), COMP_ANTI(0), COMP_VIGILANTE(0), COMP_TARJETAS(0), COMPARACION(0)) <...> BASE_ENEMIGA(Madera(0), Metal
(0), Piedra(0), Bombas(0), Tarjetas(0), AntiBombas(0), Vr(4), Sr(43), NEC_MADERA(3), NEC_METAL(3), NEC_PIEDRA
(3), NEC_MATERIAL(9), COMP_MADERA(0), COMP_PIEDRA(0), COMP_METAL(0), COMP_ROBOT(0), COMP_BOMBAS(0), COMP_ANTI
(0), COMP_VIGILANTE(0), COMP_TARJETAS(0), COMPARACION(0))
ACCION: IR_POR_TARJETA(SOLDADO)
EQUIPO_ACTIVIDADES: SOLDADO: YENDO_POR_TARJETA, BOMBER: ESPERANDO_ORDEN, ESPIA: ESPERANDO_ORDEN
EQUIPO_ENEMIGO_ACTIVIDADES: SOL_ENE: ESPERANDO_ORDEN, BOM_ENE: ESPERANDO_ORDEN, ESP_ENE: ESPERANDO_ORDEN
=====
TODOS POR UN SOLO MATERIAL
=====
<
CONTEXTO_SOLDADO(Pos(375,264), Bomber(55,35): 393, Espia(70,35): 381, Sol_Ene(920,475): 584, Bom_Ene(935,475):
598, Esp_Ene(950,475): 612, Tornado(561,286): 187, Carta(No): , Guardian(No): , Guar_Ene(No): , Robot(No):
, Robot_Ene(No): , Trampa(10,10): 443, Trampa_Ene(1000,550): 687, Bomba(No): , Refugio(No): , Ref_Ene(No): , Mad_1
(950,70): 588, Mad_2(100,500): 362, Met_1(460,280): 86, Met_2(520,240): 147, Pie_1(490,55): 239, Pie_2(510,415):
203, Base(395) Base_Ene(598) Tsole(S1,1,584) Tbone(S1,1,598) Tespe(S1,1,612) Ttor(S1,2,187) Tcarta(No,0,0)
Tbone(No,0,0) Tguare(No,0,0) Ttrae(S1,1,687) Trefe(No,0,0) Tmad1(1,588) Tmad2(1,362) Tmet1(3,86) Tmet2
(2,147) Tpie1(2,239) Tpie2(2,203) TENSIONES(Vr: 1, Sr: 17)
xxxxxxxxxx
EDO_SOLDADO(resist: 5, Nadie me ataca, No ataco, Trampa: iLibre!, Guard: iLibre!, DESPLAZANDOSE
(162,54), Desarmado)
BASE(Madera(0), Metal(0), Piedra(0), Bombas(0), Tarjetas(1), AntiBombas(0), Vr(4), Sr(49), NEC_MADERA(3), NEC_METAL
(3), NEC_PIEDRA(3), NEC_MATERIAL(9), COMP_MADERA(0), COMP_PIEDRA(0), COMP_METAL(0), COMP_ROBOT(0), COMP_BOMBAS
(0), COMP_ANTI(0), COMP_VIGILANTE(0), COMP_TARJETAS(0), COMPARACION(0)) <...> BASE_ENEMIGA(Madera(0), Metal
(0), Piedra(0), Bombas(0), Tarjetas(0), AntiBombas(0), Vr(4), Sr(43), NEC_MADERA(3), NEC_METAL(3), NEC_PIEDRA
(3), NEC_MATERIAL(9), COMP_MADERA(0), COMP_PIEDRA(0), COMP_METAL(0), COMP_ROBOT(0), COMP_BOMBAS(0), COMP_ANTI
(0), COMP_VIGILANTE(0), COMP_TARJETAS(3), COMPARACION(0))
ACCION: DESPLAZARSE(SOLDADO:162,54)
EQUIPO_ACTIVIDADES: SOLDADO: DESPLAZANDOSE(162,54), BOMBER: ESPERANDO_ORDEN, ESPIA: ESPERANDO_ORDEN
EQUIPO_ENEMIGO_ACTIVIDADES: SOL_ENE: ESPERANDO_ORDEN, BOM_ENE: ESPERANDO_ORDEN, ESP_ENE: ESPERANDO_ORDEN
=====
TODOS POR UN SOLO MATERIAL
=====
```

Fig. 3.8 Archivo de experiencia previa

Cada uno de los registros hechos son delimitados por los símbolos “>” y “<” y los comentarios hechos se encierran entre líneas de asteriscos.

Los comentarios son agregados en tiempo de juego mediante una interfaz grafica que se muestra enseguida:

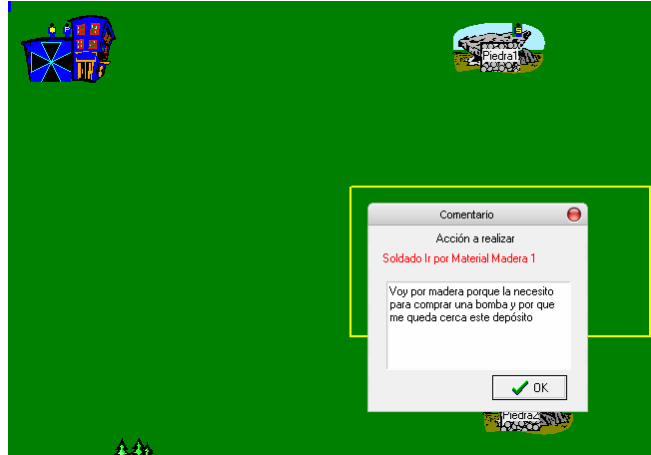


Fig. 3.9 Interfaz para la captura de un comentario en tiempo de juego

Cuando se hace un comentario, el juego se pone en pausa dándole el tiempo necesario al usuario para escribir lo que para él es importante a la hora de tomar la decisión. Debe decirse que los comentarios son muy variados y en ocasiones subjetivos. Sin embargo, se hizo un estudio de todos ellos para poderlos estandarizar y abstraer en datos que representen el comentario hecho. Eso se explicara en breve.

3.8.2 Etapa 2: Procesamiento de la experiencia Previa

Como se ha dicho anteriormente, los comentarios de los jugadores pudieran aparentar ser en ocasiones subjetivos y por lo tanto difíciles de abstraer en datos que puedan ser interpretados por la computadora.

Del estudio hecho de los comentarios se concluyó que puede hacerse una separación y especificación tanto de todos los componentes que integran una estrategia específicamente para este juego, como del plan a seguir para alcanzarla y que ya se han explicado anteriormente. Pero hay que agregar que además de ser correcta y útil la abstracción adoptada, también se tuvo que especificar las “causas” por las que se tomaba cada una de las decisiones por parte de los jugadores. Todo esto desembocó en un programa que a manera de utilidad de este proyecto que es usado por separado para hacer el procesamiento o estandarización de la experiencia previa.

En este programa se carga un archivo de una partida en una interfaz que permite seleccionar jugada por jugada y guiándose por los comentarios hechos estandarizarlos mediante una interfaz de interacción con el usuario. Este se ve así:

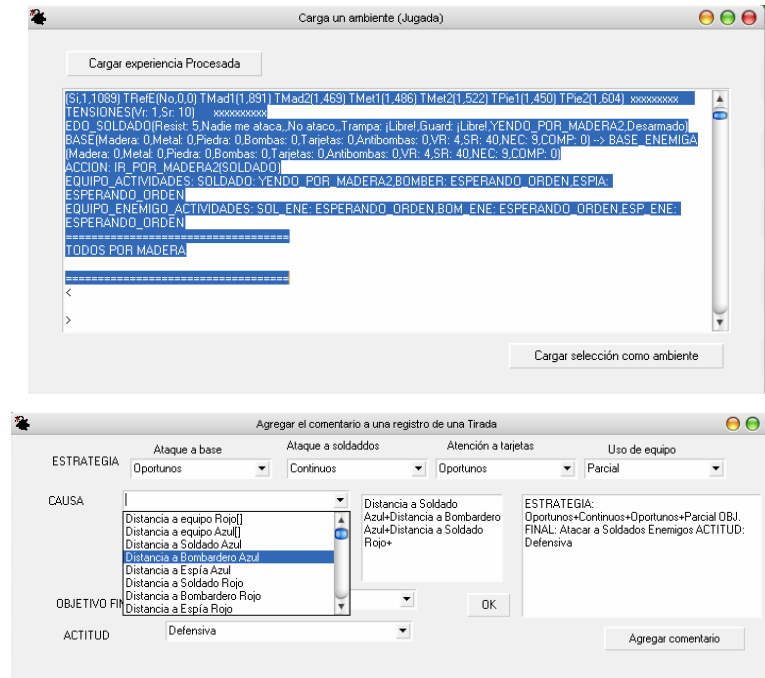


Fig. 3.10 Interfaz del programa “ProcesaExperiencia”

Una vez estandarizado el comentario mediante este programa, entonces se agrega a un nuevo archivo que lleva el mismo nombre que el original, pero con la terminación “PROC” (relativo a la palabra “Procesado”).

Un archivo de experiencia previa se ve como sigue:

```
>
CONTEXTO_BOMBER(Pos(432,295),Soldado(102,287): 330,Espia(42,152): 415,Sol_Ene(805,440): 400,Bom_Ene(926,196): 504,Esp_Ene(927,95):
534,Tornado(479,277): 50,Carta(NO): ,Guardian(NO): ,Guar_Ene(NO): ,Robot(NO): ,Robot_Ene(NO): ,Trampa(10,10): 509,Trampa_Ene(1000,550):
623,Bomba(NO): ,Refugio(NO): ,Ref_Ene(NO): ,Mad_1(930,70): 546,Mad_2(100,500): 390,Met_1(460,280): 32,Met_2(520,240): 104,Pie_1(490,55)
247,Pie_2(510,415): 143,Base(40,55): 460,Base_Ene(925,500): 534) TSoE(S1,1,400) TBOM(E(S1,1,504) TESPE(S1,1,534) TTOR(S1,3,50) TCarta
(NO,0,0) TBomba(NO,0,0) TGuarE(NO,0,0) TTraE(S1,1,623) TRefE(NO,0,0) TMad1(1,546) TMad2(1,390) TMet1(3,32) Tmet2(2,104) TPie1(2,247) TP
(2,143) xxxxxxxx TENSIONES(Vr: 1, Sr: 18) xxxxxxxx
EDO_BOMBER(Resist: 5,Nadie me ataca,,No ataco,,Trampa: ,Libre!,Guard: ,Libre!,YENDO_POR_MADERA2,Desarmado)
BASE[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(40),NEC_MADERA(3),NEC_METAL(3),NEC_PIEDRA(3),NEC_ENEMIG
[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(40),NEC(3,3,3):9,COMP(0,0,0,0,0,0): 0]
ACCION: DESPLAZARSE(BOMBER,234,446)
EQUIPO_ACTIVIDADES: SOLDADO: YENDO_POR_MADERA2,BOMBER: DESPLAZANDOSE(234,446),ESPIA: YENDO_POR_MADERA2
EQUIPO_ENEMIGO_ACTIVIDADES: SOL_ENE: YENDO_POR_PIEDRA2,BOM_ENE: YENDO_POR_MADERA1,ESP_ENE: YENDO_POR_MADERA1
=====
POR LO DE SIEMPRE ME VOY DE AQUI
ESTRATEGIA: Continuos + Nunca + Oportunos + Total CAUSA: Distancia a tornado + Distancia a equipo Rojo[] + Distancia a Materiales[] +
FINAL: Bombardear base ACTITUD: Ofensiva
=====
<
```

Fig. 3.11 Ejemplo de un registro procesado por el programa “ProcesaExperiencia”.

Debe hacerse notar que muchos de los comentarios se estandarizaron después de hacer preguntas específicas y complementarias a los usuarios al y tomar notas de lo que ellos no apuntaron en la interfaz destinada para eso en el juego.

3.8.3 Etapa 3: Implementación de la Adaptación del Modelo del Proceso Creativo.

La implementación de este se puede explicar gráficamente con la siguiente ilustración:

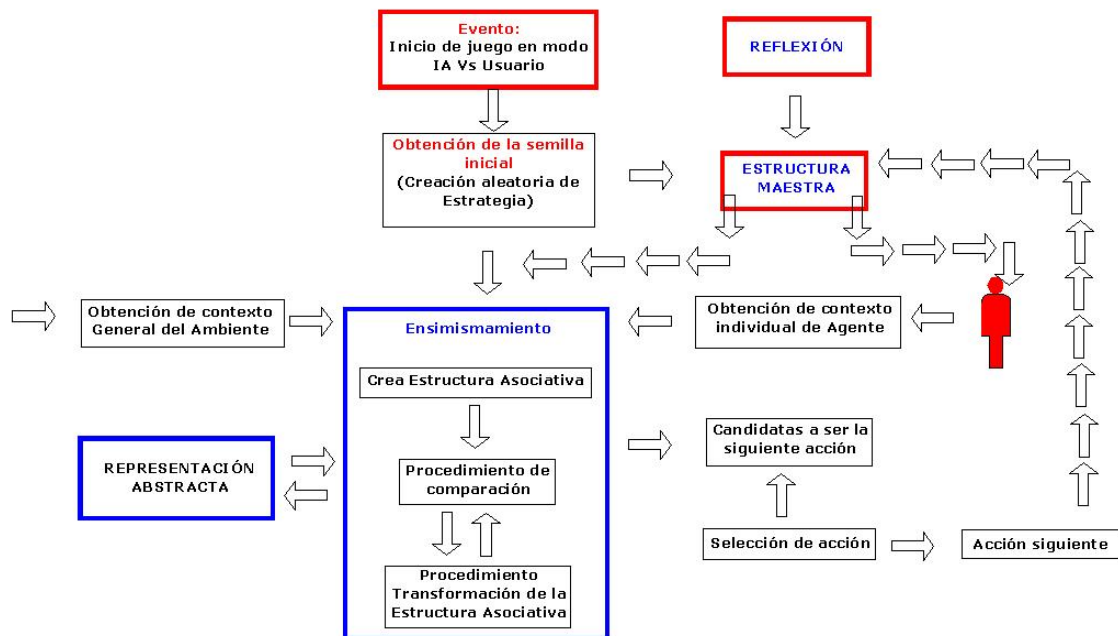


Figura 11. Esquema general del funcionamiento de la adaptación del modelo Ensimismamiento-Reflexión.

El modelo Ensimismamiento-Reflexión establece que, para iniciar su funcionamiento se requiere de una “semilla” inicial la cual es definida por el usuario.

En la adaptación del modelo al videojuego para la generación de estrategias la semilla inicial esta dada por las condiciones generales del juego.

En general, la adaptación esta guiada hacia la reacción del modelo de acuerdo a las condiciones dominantes en un instante dado, pero respetando la especificación de le estrategia.

El estado general del juego es descrito por el valor de todas las variables involucradas. Esto quiere decir que para que el funcionamiento del modelo sea correcto se debe de tener todas estas variables especificadas para que la implementación de la Inteligencia Artificial tenga “conciencia” (tenga acceso a todos los datos necesarios) del estado completo del juego para poder tomar una decisión.

3.8.3.1 Al Inicio del Juego

Pero puesto que al inicio del juego las variables que describen el ambiente del juego tienen valores nulos o de cero, entonces se han implementado dos formas distintas de poder definir los datos iniciales para proporcionarle al modelo la semilla inicial para su correcta .

1. Elegir aleatoriamente uno a uno cada elemento de los que describen una situación y modo de juego por parte del equipo (Completamente aleatoria)
2. Elegir aleatoriamente una descripción y modo de juego de la experiencia previamente procesada o Representación Abstracta (Aleatoria sobre Estructura asociativa)

Estas opciones tienen un funcionamiento particular, pero en cualquiera de ellas los ciclos característicos entre los estados el modelo Ensimismamiento-Reflexión, entran en acción.

3.8.3.1.1 Inicio Completamente Aleatorio

Al inicio del juego, los elementos que pueden ser elegidos aleatoriamente son:

1. Estrategia
2. Objetivo final
3. Actitud
- 4.

Esto se hace mediante la generación aleatoria de números la cual da por resultado una opción para cada uno de los anteriores elementos y cada uno de sus componentes.

3.8.3.1.2 Inicio Aleatorio sobre Representación Abstracta

Esta se lleva a cabo mediante una función generadora de números aleatorios cuyo resultado corresponde al número de registro en la Representación Abstracta y de donde se traen los valores de todos los parámetros necesarios para crear la semilla inicial de juego.

3.9 Esquema general del funcionamiento de la adaptación del modelo

El funcionamiento general del modelo Ensimismamiento-Reflexión se puede ejemplificar con la siguiente ilustración.

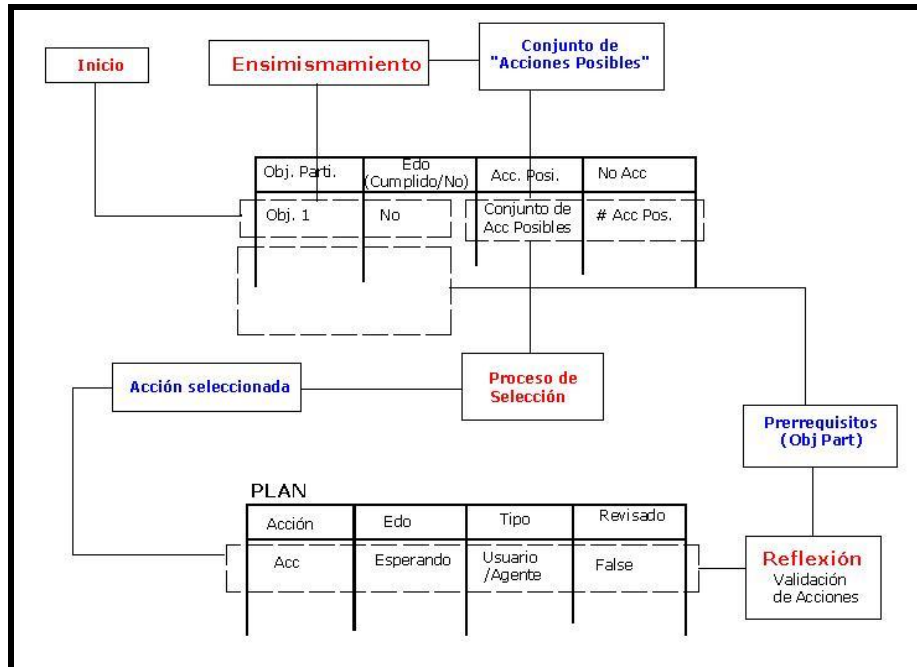


Fig. 3.14 Esquema general del funcionamiento del modelo Ensimismamiento-Reflexión

Para que pueda funcionar correctamente el modelo se supone se tiene acceso a todas las variables que especifican el ambiente predominante en el juego en el tiempo actual. Si se trata del inicio del juego se debe haber ejecutado ya la selección inicial que se explicó en la sección anterior. Si es en la caso de tiempo de juego se da por entendido que todas estas ya están definidas y se tiene acceso a ellas.

En general, la intervención de la implementación del modelo se puede ver como un algoritmo paso a paso, el cual se puede explicar como sigue:

1. Obtención de los datos del ambiente general actual o semilla inicial. (En el caso de encontrarse al inicio del juego, estos son obtenidos mediante cualquiera de las selecciones aleatorias o mediante el muestreo y obtención de los datos con su valor real actual, sin importar que estos sean nulos o iguales a cero).
2. Construcción de la “Estructura Asociativa”. Esto es, la creación de una estructura que sigue la misma platilla que un átomo (abstracción de la información que describe completamente el ambiente) pero estrictamente

dinámica, pues debe ser susceptible a transformaciones para el funcionamiento correcto del modelo.

3. Estado de ensimismamiento. Con la estructura Asociativa creada en el paso anterior se hace una comparación con toda la representación abstracta, registro por registro para encontrar las acciones candidatas a ser la siguiente acción. Debido al gran dinamismo del ambiente, siempre hay la posibilidad de que no se encuentre una acción en la Representación Abstracta que corresponda exactamente a la estructura asociativa que se está usando para este estado. Por lo que se hace uso de transformaciones de la Estructura Asociativa para poder lograr una correspondencia y poder traer acciones candidatas. Sin embargo, a pesar de estas transformaciones todavía se considera la posibilidad de no encontrar correspondencias, a lo que se le denomina un “Estancamiento” el cual es resuelto en el siguiente estado (reflexión).
4. Estado de Reflexión. Es este estado se revisa el material encontrado durante el estado anterior. Se revisa y se valida para ver su utilidad dentro del contexto predominante en este momento exacto, así como su validez para seguir con la continuidad y lógica del juego respetando la estrategia de juego. De entre la interacción entre el estado Ensimismamiento y reflexión se obtiene el plan a seguir para alcanzar el objetivo final.
5. Asignación de tareas. Mediante una constante serie de actividades o acciones cognitivas (acciones de evaluación) se asignan las tareas a cada uno de los agentes activos (vivos) de acuerdo a la conveniencia o grado de ventaja de cada uno de los agentes y considerando sus habilidades particulares. Finalmente se ejecutan.

Todos los anteriores procesos se explican en lo siguiente.

3.10 Obtención de los datos del ambiente general actual o semilla inicial

Como se indicó en la sección de inicio aleatorio. La obtención de los datos puede ser de manera aleatoria de dos maneras distintas. En la primera se hace una

elección totalmente aleatoria de todos los aspectos y variables necesarias para el funcionamiento del modelo. Esto se hace mediante funciones de generación aleatoria. La implementación de una de estas se ve como sigue:

```

////////////////////////////////////
procedure EligeEstrategiaRandom;
var
  opcion : integer;
begin
  InicializaEstructuraMaestra;
  // 0 = nunca; 1= Continuos; 2= Oportunos
  // ataques a base
  randomize;
  opcion := random(3); // 0 <= opcion < 3 [0,1,2]
  case opcion of
    0 : EstructuraMaestra.Estrategia[1] := 'Nunca';
    1 : EstructuraMaestra.Estrategia[1] := 'Continuos';
    2 : EstructuraMaestra.Estrategia[1] := 'Oportunos';
  end;

  //ataques a soldados
  randomize;
  opcion := random(3);
  case opcion of
    0 : EstructuraMaestra.Estrategia[2] := 'Nunca';
    1 : EstructuraMaestra.Estrategia[2] := 'Continuos';
    2 : EstructuraMaestra.Estrategia[2] := 'Oportunos';
  end;

  //Atención a tarjetas
  randomize;
  opcion := random(3);
  case opcion of
    0 : EstructuraMaestra.Estrategia[3] := 'Nunca';
    1 : EstructuraMaestra.Estrategia[3] := 'Continuos';
    2 : EstructuraMaestra.Estrategia[3] := 'Oportunos';
  end;

  //Uso del equipo
  randomize;
  opcion := random(2);
  case opcion of
    0 : EstructuraMaestra.Estrategia[4] := 'Total';
    1 : EstructuraMaestra.Estrategia[4] := 'Parcial';
  end;
end;
////////////////////////////////////

```

Fig. 3.15 Implementación de la generación aleatoria de la semilla inicial para el funcionamiento del modelo Ensimismamiento-Reflexión.

Cada uno de los elementos, como se ve en la ilustración anterior, es elegido aleatoriamente. Todas las demás funciones tienen una estructura y lógica similar.

En el caso de la selección aleatoria sobre la Representación abstracta, su implementación se ve como sigue:

```

////////////////////////////////////
// carga la Estrategia y el Obj Final ademas de determinar los Particu
procedure CargaEstraYObjFinalDeRepAbstracta;
var
  direc : string;
  opi : integer;
  noAtomos : integer;
  linea : string;
  estrategia : UEnsimismamiento.contElemEstrat;
begin
  // determina el numero de atomos en la representación Abstracta
  noAtomos := 0;
  direc := UTablero.Directorio;
  direc := direc + '\R_ABSTRACTA.txt';
  AssignFile(arch,direc);
  reset(arch);
  While not EOF(arch) do begin
    readln(arch, linea);
    if Pos('ATOMO', linea) > 0 then noAtomos := noAtomos + 1;
  end;
  CloseFile(arch);

  // elige aleatoriamente el atomo para cargar estrategia y ObjFinal
  randomize;
  opi := random(noAtomos + 1);

  // adquiere la estrategia
  linea := DameEstrategiaDeAtomo(opi);
  // carga la estrategia en la estructura Maestra
  estrategia := UEnsimismamiento.SeparaEstrategia(linea);
  EstructuraMaestra.Estrategia[1] := estrategia[1];
  EstructuraMaestra.Estrategia[2] := estrategia[2];
  EstructuraMaestra.Estrategia[3] := estrategia[3];
  EstructuraMaestra.Estrategia[4] := estrategia[4];

  //adquiere el ObjFinal
  linea := DameObjFinalDeAtomo(opi);
  // carga el ObjFinal;
  EstructuraMaestra.ObjFinal := linea;

```

Fig. 3.16 Implementación de la generación de la representación Abstracta.

En donde se puede notar que se hace un contero del número de registros en la Representación Abstracta, después de lo cual se ejecuta una generación aleatoria de números que corresponderá al número de registro o átomo registrado del cual se toman los datos necesarios para generar la semilla inicial. Teniendo completos estos se entra al estado de ensimismamiento.

3.11 Estado de Ensimismamiento

En general se trata de encontrar en la Representación Abstracta de la experiencia previa un registro igual o lo mas parecido posible a las circunstancias presentes y traer consigo la acción relacionada a ese registro y considerarla como una de las acciones candidatas a ser la siguiente.

La comparación se hace de dos modos:

1. Comparación Exacta. Con el fin de traer una acción que corresponda exactamente a la situación presente.
2. Comparación bajo transformación. Se trata de dar de baja datos que no son trascendentes para la situación actual y encontrar un registro en la representación Abstracta que corresponda con esa transformación.

De todas las acciones encontradas se hace un arreglo y se ponen a disposición del proceso de toma de decisión en donde se elige la acción.

Inicialmente se toma una “fotografía instantánea” de la situación en el tablero de juego, esto es, se toman los datos de los agentes activos, su posición, su estado (nivel de resistencia), sus niveles de tensión (valores numéricos que representan su grado de vida y seguridad en riesgo. La seguridad en riesgo está determinada como la proximidad del agente en cuestión con relación a los enemigos y su vida en riesgo por su nivel de resistencia), su actividad en este momento, el número de unidades de material almacenadas en la base de control, entre muchos otros datos.

Se hace una de estas fotografías por agente activo en el equipo, y puede tener la siguiente forma:

```

CONTEXTO_SOLDADO(Pos(40,35),Bomber(55,35): 15,Es pia(70,35): 30,Sol_Ene(920,475): 984,Bom_Ene
(935,475): 997,Esp_Ene(950,475): 1011,Tornado(NO): ,Carta(NO): ,Guardian(NO): ,Guar_Ene(NO):
,Robot(NO): ,Robot_Ene(NO): ,Trampa(10,10): 39,Trampa_Ene(1000,550): 1089,Bomba(NO): ,Refugio
(NO): ,Ref_Ene(NO): ,Mad_1(930,70): 891,Mad_2(100,500): 469,Met_1(460,280): 486,Met_2(520,240)
522,Pie_1(490,55): 450,Pie_2(510,415): 604,Base(40,55): 20,Base_Ene(925,500): 1000) TSole
(1,1,984) TBome(Si,1,997) TEspe(Si,1,1011) TTor(NO,0,0) TCarta(NO,0,0) TBomba(NO,0,0) TGuarE
(NO,0,0) TTrAE(Si,1,1089) TRefe(NO,0,0) TMad1(1,891) TMad2(1,469) TMet1(1,486) TMet2(1,522)
TPie1(1,450) TPie2(1,604) xxxxxxxxxx TENSIONES(Vr: 1, Sr: 10) xxxxxxxxxx
EDO_SOLDADO(Resist: 5,Nadie me ataca ,No ataco ,Trampa: ¡Libre!,Guard: ¡
Libre!,YENDO_POR_MADERA2,desarmado)
BASE[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(40),NEC_MADERA
(3),NEC_METAL(3),NEC_PIEDRA(3),NECE_ENEMIGA[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas
(0),Antibombas(0),VR(4),SR(40),NEC(3,3,3):9,COMP(0,0,0,0,0,0,0,0)]
ACCION: IR_POR_MADERA2(SOLDADO)
EQUIPO_ACTIVIDADES: SOLDADO: YENDO_POR_MADERA2,BOMBER: ESPERANDO_ORDEN,ESPIA: ESPERANDO_ORDEN
EQUIPO_ENEMIGO_ACTIVIDADES: SOL_ENE: ESPERANDO_ORDEN,BOM_ENE: ESPERANDO_ORDEN,ESP_ENE:
ESPERANDO_ORDEN

```

Fig. 3.17. Ejemplo de captura datos en tiempo de juego

Una vez tomadas estas, se hace una estructura asociativa por cada uno de los agentes, que como ya se explicó es una estructura dinámica con la plantilla de un átomo, que a su vez, es la abstracción de la información anteriormente obtenida. Una Estructura asociativa se vería como sigue:

```

CONTEXTO_G: BASE(0,0,0,0,0,0,0,4,43,3,3,3,9,0,0,0,0,0,0,0,0,0) BASE_ENEMIGA
(0,0,0,0,0,0,4,43,3,3,3,9,0,0,0,0,0,0,0,0,0,0)
CONTEXTO_P: [(13,287,63,87),1,NO,NO,20,1000,1,1,1,NO,NO,NO,1,NO,1,1,1,1,1,1,11]
Estrategia: Continuos+Nunca+Oportunos+Total
Causa: TCarta(Si,1,404)DCarta(13,287,63,87)TSole(Si,1,984)TBome(Si,1,997)TESpe(Si,1,1011)
Obj. Final: Bombardear base
Actitud: ofensiva

```

Fig. 3. 18 *Abstracción de la información como estructura Asociativa*

La construcción de estas se puede ver mediante un procedimiento que se ha implementado como sigue:

En donde se aprecia que se toman los datos generales del ambiente para cada uno de los agentes, su contexto particular y todos los elementos que dirigen la generación de la estrategia de juego.

Una vez construías las Estructuras Asociativas se entra en el proceso de comparación entre cada una de estas y el Representación Abstracta. Primero en el modo Exacto y si no se tienen resultados entonces se entra en el proceso de transformación de las Estructuras Asociativas para hacer las comparaciones.

Para poder explicar las transformaciones se ha hecho un archivo paso a paso del proceso de ensimismamiento y que se explica a continuación brevemente.

Las transformaciones hechas a las Estructuras asociativas consisten en la eliminación de información que no es trascendente para hacer una comparación. Como se ve en el siguiente ejemplo:

Teniendo la siguiente Estructura Asociativa:

```

----- ESTRUCTURA ASOCIATIVA
BASE_ENEMIGA(0,0,0,0,0,0,4,40,3,3,3,9,0,0,0,0,3,0,0,0,3)
[NO,NO,NO,NO,975,25,1,1,1,NO,NO,NO,1,NO,1,1,1,1,1,1,10]
Oportunos+Nunca+Oportunos+Parcial
Acumular
Defensiva

```

Fig. 3.19 Ejemplo de una Estructura Asociativa

Se compara con uno de los átomos registrados en la Representación Abstracta, digamos... el siguiente:

```

----- ATOMO
ATOMO #0
CONTEXTO_G: BASE(0,0,0,0,0,0,4,43,3,3,3,9,0,0,0,0,0,0,0,0) BASE_ENEMIGA
(0,0,0,0,0,0,4,43,3,3,3,9,0,0,0,0,0,0,0,0)
CONTEXTO_P: [(13,287,63,87),1,NO,NO,20,1000,1,1,1,NO,NO,NO,1,NO,1,1,1,1,1,11]
Estrategia: Continuos+Nunca+Oportunos+Total
Obj. Final: Bombardear base
Actitud: ofensiva
Acción: IR_POR_TARJETA(SOLDADO)

```

Fig. 3.20 Ejemplo de átomo de comparación

Al intentar una comparación exacta se tienen los siguientes resultados:

----- TRANSFORMACIONES -----

Contexto Gral. sin transformación-->Comparación exacta
Contexto Part. sin transformación-->Comparación exacta
Estrategia: Ataque a base, a soldados y atención a tarjetas

----- CONTEXTOS GENERALES COMPARADOS -----

JUEGO: BASE_ENEMIGA(0,0,0,0,0,0,4,40,3,3,3,9,0,0,0,0,3,0,0,0,3)
ABSTR: BASE(0,0,0,0,0,0,4,43,3,3,3,9,0,0,0,0,0,0,0,0)

Comparando elementos 1 a 21...

Elem: 1(0 con 0):-1
Elem: 2(0 con 0):-1
Elem: 3(0 con 0):-1
Elem: 4(0 con 0):-1
Elem: 5(0 con 0):-1
Elem: 6(0 con 0):-1
Elem: 7(4 con 4):1
Elem: 8(40 con 43):0

----- CONTEXTOS PARTICULARES COMPARADOS -----

JUEGO: [NO,NO,NO,NO,975,25,1,1,1,NO,NO,NO,1,NO,1,1,1,1,1,1,10]
ABSTR: CONTEXTO_P: [(13,287,63,87),1,NO,NO,20,1000,1,1,1,NO,NO,NO,1,NO,1,1,1,1,1,1,11]

Comparando elementos 1 a 22...

Elem: 1(NO con (13)):0

----- ESTRATEGIAS COMPARADAS -----

JUEGO: Oportunos+Nunca+Oportunos+Parcial
ABSTR: Continuos+Nunca+Oportunos+Total
(Oportunos-->Continuos)
(Nunca-->Nunca)
(Oportunos-->Oportunos):0

----- OBJETIVOS COMPARADOS -----

JUEGO: Acumular
ABSTR: Obj. Final: Bombardear base
(Acumular con Bombardear base):0

----- ACTITUDES COMPARADOS -----

JUEGO: Defensiva
ABSTR: Actitud: ofensiva
(Defensiva con ofensiva):0

Falla de comparación

Fig. 3.21 Seguimiento de comparación y transformación entre Estructura Asociativa y átomos en la Representación Abstracta del conocimiento.

Pero que al ser transformada se tienen coincidencias.

Las transformaciones son de dos tipos:

1. Definición de tolerancias de comparación entre los datos comparados (80%)
2. Eliminación de datos (filtros).

3.11.1 Definición de tolerancias de comparación entre los datos comparados

Por ejemplo, al darle tolerancias del 80% de semejanza a los datos de las tensiones totales generales (suma total de las tensiones de vida y seguridad en riesgo) y extirpar el dato de “Uso de equipo” de la estrategia (pues el que un jugador use o no todo el equipo en una sola actividad no es trascendente para hacer una comparación) y la atención a las tarjetas de poder (que aunque puede influenciar en la toma de una decisión no es trascendente para una comparación), entonces se obtiene una comparación y por lo tanto una acción candidata a ser la siguiente.

Hablar de una tolerancia del 80% de semejanza significa que, si tengo una tensión de un valor total de 100 (solo para efectos de un ejemplo) registrada en la Representación Abstracta, entonces la consideraré “igual o semejante” a la de la Estructura Asociativa en cuestión si, y solo si esta tiene un valor para la misma tensión en un rango entre 80 y 100.

Si esta condición se da y la comparación da banderas positivas con respecto al criterio tomado, entonces se obtiene la acción asociada al átomo que es objeto de comparación.

El proceso de Ensimismamiento también incluye la selección de la acción del conjunto de acciones obtenido en el paso anterior.

Si el resultado de la comparación es una sola acción o si se obtiene una acción producto de una comparación exacta entonces se pasa inmediatamente al siguiente estado tomando como base esa acción.

En cambio, si son más de una acción las producidas por las comparaciones, entonces se escoge una de las acciones mediante dos procesos diferentes:

1. Aleatoriamente. Se elige una de las acciones del conjunto de candidatas obtenidas en el proceso de comparación.

2. Se elige por prioridad. De acuerdo a la transformación hecha, se tienen niveles de prioridad. Siendo el nivel mayor de prioridad la comparación sin transformaciones.

La acción tomada se agrega al plan general y se entra al estado de Reflexión.

3.11.2 Eliminación de datos (filtros)

Como ya se ha mencionado, la intervención de la implementación de la adaptación del modelo E-R es activada por eventos que suceden en el ambiente de juego o por activación debida a las acciones cognitivas por parte de los agentes que componen el equipo controlado por la IA, o por las acciones cognitivas de evaluación a nivel ambiente de juego.

Cuando la IA entra en acción se forman las Estructuras Asociativas de cada uno de los agentes del equipo, las cuales son transformadas mediante los filtros de información para que contengan los datos estrictamente necesarios para que los estados Ensimismamiento y Reflexión puedan ejecutar su trabajo.

El filtrado de la información depende del evento, es decir, que dependiendo de lo que active la intervención del modelo es la información usada para su correcta ejecución.

El filtrado se hace de la siguiente manera:

Habiéndose dado la bandera que activa la intervención de la implementación del modelo, el evento que dio esta señal, activa el agregado de información a la EA de cada uno de los agentes. Por ejemplo, si el evento es “Apareció un Tornado” entonces, al final de las EA, se agrega la marca: “TTor” que es interpretada como la Tensión debida a la Aparición de un tornado. Este agregado de información se considera en este trabajo como una primera transformación de una EA.

Cada uno de los eventos agrega de esta forma su marca característica a la EA con la cual va a trabajar el modelo.

Una vez agregada esta marca, el estado Ensimismamiento empieza a hacer comparaciones en busca de todos aquellos átomos contenidos en la Representación Abstracta del conocimiento que contengan en su sección “Causas” la marca que se agregó en el paso anterior a las EA.

Una vez obtenidos todos los átomos que corresponden a este criterio de comparación se efectúa una segunda transformación.

De los átomos encontrados en el paso anterior, se extraen todas las “causas” contenidas. Estas causas sirven como filtro de las EA’s, de modo que de las EA’s se extraen los datos correspondientes a las causas registradas en los átomos obtenidos en el paso anterior.

De todos estos datos extraídos de las EA’s de los agentes y de los contenidos en la sección de causas de los átomos obtenidos se hace una comparación. Si existe una correspondencia según los criterios establecidos entre estos datos, entonces se extraen las acciones registradas en los átomos en los que se presentó la correspondencia, y se consideran a estas como las posibles siguientes acciones.

En el caso de que no exista una correspondencia de datos, entonces se procede a hacer una segunda búsqueda.

Con los datos extraídos de las EA’s de los agentes y que corresponden a las causas registradas en los átomos obtenidos en la primer búsqueda, se empieza a hacer comparaciones con todos los átomos contenidos en la Representación Abstracta del conocimiento, tengan o no en su sección de causas, la marca agregada a las EA’s debidas al evento que activó este proceso.

Si se encuentra alguna correspondencia de acuerdo a los criterios establecidos entonces, se extraen las acciones de los átomos con los que se presentó dicha correspondencia y se considera a estas como las posibles siguientes acciones.

3. 12 El Estado Reflexión

En este estado se revisa el plan, es decir, se compara cada una de las acciones en el plan y se verifica su validez y continuidad del plan. Esto es, dada una acción se verifican sus prerrequisitos para ser ejecutada. Si estos se cumplen entonces se da por válida la acción, sino es así entonces los prerrequisitos se agregan a las metas particulares del plan, es decir, se especifica lo necesario que debe existir para poder ejecutar esa acción.

Se hace entonces una revisión de todas aquellas acciones en la lista de acciones primitivas tales que tengan por poscondiciones las precondiciones requeridas o como se le llaman en este proyecto: “Objetivos particulares” mismas que se agregan al plan.

Por cada cambio en el plan se hace la misma revisión, alternando entre el estado de Ensimismamiento y Reflexión para traer las acciones necesarias al plan para cumplir todos los objetivos particulares los cuales llevan al objetivo final.

Por ejemplo:

Si tengo la siguiente especificación de estrategia:

Estrategia	
Ataque Base enemiga	Continuos
Ataque soldados enemigos	Nunca
Atención tarjetas de poder	Oportunos
Uso del equipo	Total
Objetivo final	
Objetivo Final	Bombardera Base
Actitud	
Actitud	Ofensiva

Fig. 3.22 Ejemplo de especificación de parámetros de estrategia de Juego

Por la intervención del estado de Ensimismamiento se puede traer como Objetivo Particular “Bombas(1)” puesto que el objetivo es “Bombardera Base” para lo que se necesita tener por lo menos 1 bomba almacenada.

El proceso de Reflexión entonces revisaría que necesito para tener una bomba almacenada, de lo que puede resultar: “Comprar bomba”, lo cual se agrega al plan. Puesto que hubo un supuesto cambio en plan, una vez más entra el estado de reflexión en acción y revisa la validez de todas las acciones contenidas en el plan.

En este caso teórico nota que para “Comprar bomba” requiero tener cierto número de unidades de material almacenado como precondition.

Como esta condición no existe (en este ejemplo) entonces se traen las acciones que solucionan el problema, y entonces puede obtenerse: Ir por madera, Ir por Metal o comprarlo e Ir por Piedra como acciones que se agregan al plan.

Como hubo un cambio se revisa una vez mas, hasta que todo esta lógicamente establecido y especificado.

3. 13 La Asignación de Tareas

Cuando el plan está completo (cuando la bandera de validez de reflexión es positiva) entonces se asignan las actividades a cada uno de los agentes activos en el equipo tomando en cuenta el elemento “uso de equipo” de la estrategia de juego. Eso se hace mediante una acción cognitiva de evaluación la cual prueba a cada uno de los agentes y determina el grado de peligrosidad personal para ejecutar una acción determinada dentro del plan y tomando en cuenta sus habilidades personales, pues no todos pueden hacer todas las actividades que pueden tener lugar en el juego.

El grado de peligrosidad y la especificación e la “actitud de juego” influyen sobre la asignación de las tareas. Mientras que una actitud defensiva o prudente le da mucha importancia al valor numérico que representa el nivel de peligrosidad de ejecutar una acción en particular por un agente en particular, una actitud agresiva no le dará tanta importancia.

Tomando en cuenta estos criterios se hace la asignación de las tareas y se ejecutan dando como resultado el movimiento de los agentes en el tablero de juego.

3.14 Las Acciones Cognitivas

En general, hay dos acciones cognitivas o de evaluación

1. Asignación de tareas (Estado: Realizado). (que revisa los parámetros de especificación de la estrategia y la conveniencia para cada uno de los agentes activos para ejecutar una acción)
2. Evaluación de peligros. (Estado: En proceso de Implementación) En cada movimiento cada agente debe estar consciente de sus circunstancias personales, pues debe de “Saber” cuando una trampa o una bomba aparece. Sin embargo, es la intervención del modelo la que encuentra la solución del problema “Si lo hay” pero activada por la acción cognitiva de evaluación. A semejanza del vivir cotidiano de una persona, es como encontrarse en una situación cuya evaluación le dice “Tienes un problema enfrente” y entonces esta persona pregunta: “¿Qué hago?” Quien da el anuncio del evento es el proceso de evaluación, mientras que el modelo intenta encontrar la respuesta a la pregunta “¿Qué hago?”

3.15 Los Estancamientos

El modelo Ensimismamiento-Reflexión establece una forma particular de definir y tratar con un estancamiento el cual puede tener lugar durante el estado de Ensimismamiento y que también ha sido adaptado a este proyecto.

En este contexto, se define un estancamiento el estado generado cuando, durante el estado de Ensimismamiento, el contexto general, al ser comparado con la Estructura Abstracta en busca de la(s) siguiente(s) posible(s) acción(es) no se encuentra ninguna coincidencia, por lo que no se generan acciones para seguir trabajando en la construcción del material actual.

Textualmente, Pérez y Pérez [7] explica cómo es tratado de la siguiente forma:

“Para ilustrar cómo MEXICA (programa de implementación del modelo Ensimismamiento-Reflexión) resuelve el problema, el lector puede imaginar a un escritor que cuando se encuentra en el desarrollote una historia llega a estar bloqueado. Para “romper” con el estancamiento, este autor hipotético puede recordar y analizar cómo otros escritores han tratado con situaciones similares y aplicar el mismo método o una variación de ésta en la historia en progreso. De la misma forma, cuando MEXICA está tratado de romper con un estancamiento, éste explora el espacio conceptual definido por la experiencia previa (en el caso específico de MEXICA, cuentos cortos) y analiza qué eventos han seguido a la acción que causó el estancamiento para obtener las la forma de salir de éste.”

En otras palabras, el método que establece Pérez y Pérez en su modelo Ensimismamiento-Reflexión para poder salir de un estancamiento es “copiando” la forma en que la acciones han sido usadas en la experiencia previa.

Este proceso consiste en obtener de la experiencia previa todas aquellas acciones que siguen a la acción que corresponde a aquella que en el material actualmente en desarrollo ocasionó el estancamiento.

Eliminar las acciones inadecuadas para romper con el estancamiento y finalmente agregarla al material actualmente desarrollándose y entra una vez más al estado de ensimismamiento, seguido del estado de reflexión para validar la acción recién agregada.

El caso particular de este proyecto, un estancamiento puede presentarse en una situación en particular, es decir, cuando en una acción cognitiva de tipo agente (cuando un agente hace una evaluación de su estado particular) y descubre que requiere de “reaccionar” para asegurar su supervivencia, pero al entrar al estado de

ensimismamiento para encontrar la posible acción que lo saque de su estado de peligro no encuentra una coincidencia en la representación abstracta de la experiencia previa.

Cuando se dan las anteriores circunstancias, se trata de romper el estancamiento de la misma manera como lo establece el modelo, se hace una revisión de la experiencia previa sin procesar (de acuerdo con el modelo original, la representación concreta del conocimiento o experiencia previa), en este caso particular los archivos obtenidos en la fase de experimentación obteniendo la acción inmediata efectuada por un agente en particular y que corresponde a la acción actual del agente en el que tuvo lugar el estado de estancamiento.

Resumen

En este capítulo se hizo la revisión detallada de la adaptación del modelo Ensimismamiento-Reflexión para que funcione como un generador de estrategias en tiempo de juego para un videojuego de estrategia llamado “Guns”. Además se mencionó y describió todos los programas que a manera de herramientas se usaron para implementar la adaptación de este modelo. Se explicó también la correspondencia entre todas las etapas del desarrollo de este trabajo y la generación de la información necesaria para hacer esta adaptación.

Capítulo 4

Cómo funciona la adaptación del modelo Ensimismamiento-Reflexión

En este capítulo se muestra la forma en que trabaja la implementación de la adaptación del modelo Ensimismamiento-Reflexión y se hace un análisis de su funcionamiento mediante un seguimiento paso a paso.

Así mismo, en este capítulo el desarrollo de la fase de experimentación en donde se adquirió la experiencia previa con la que trabaja la IA y las herramientas usadas para su procesamiento, puesto que el concepto que sustenta la transformación de la información es un resultado del análisis hecho a la adaptación del modelo usado, mismo que forma sustenta parte importante de las conclusiones hechas al respecto de esta investigación.

4.1 Obtención de la experiencia previa

Se entiende por experiencia previa al conjunto de archivos de texto que almacenan la información obtenida en la fase de experimentación y que corresponde al almacenamiento paso a paso de cada una de las partidas celebradas usando el videojuego en un enfrentamiento entre dos usuarios.

En esta fase de experimentación, participaron 5 voluntarios quienes inicialmente no tenían ninguna experiencia con el juego. En su primer contacto con éste se les dio

una explicación de la dinámica y reglas del juego, se les explicó el uso de los controles y se les pidió escribieran comentarios por cada uno de sus movimientos del porqué de sus decisiones usando para ello la interfaz diseñada para agregar texto en cada tirada.

Se obtuvieron un total de 23 partidas completas en donde hubo siempre un ganador.

Un ejemplo de un registro de una tirada hecha durante una partida de juego se ve como sigue:

```
>
  CONTEXTO_SOLDADO(Pos(40,35),Bomber(55,35): 15,Espia(70,35): 30,Sol_Ene(920,475):
984,Bom_Ene(935,475): 997,Esp_Ene(950,475): 1011,Tornado(No): ,Carta(373,263):
404,DCarta(13,287,63,87),Guardian(No): ,Guar_Ene(No): ,Robot(No): ,Robot_Ene(No):
,Trampa(10,10): 39,Trampa_Ene(1000,550): 1089,Bomba(No): ,Refugio(No): ,Ref_Ene(No):
,Mad_1(930,70): 891,Mad_2(100,500): 469,Met_1(460,280): 486,Met_2(520,240):
522,Pie_1(490,55): 450,Pie_2(510,415): 604,Base(20) Base_Ene(1000) TSole(Si,1,984)
TBomE(Si,1,997) TEspE(Si,1,1011) TTor(No,0,0) TCarta(Si,1,404) TBomba(No,0,0)
TGuarE(No,0,0) TTraE(Si,1,1089) TRefE(No,0,0) TMad1(1,891) TMad2(1,469) TMet1(1,486)
TMet2(1,522) TPie1(1,450) TPie2(1,604) xxxxxxxxxx TENSIONES(Vr: 1,Sr: 11)
xxxxxxxxxxx
  EDO_SOLDADO(Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard:
¡Libre!,YENDO_POR_TARJETA,Desarmado)
  BASE[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(43
),NEC_MADERA(3),NEC_METAL(3),NEC_PIEDRA(3),NEC_MATERIAL(9),COMP_MADERA(0),COMP_PIEDRA(0
),COMP_METAL(0),COMP_ROBOT(0),COMP_BOMBAS(0),COMP_ANTI(0),COMP_VIGILANTE(0),COMP_TARJETAS
(0),COMPARACION(0)] <...>
  BASE_ENEMIGA[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(4
3),NEC_MADERA(3),NEC_METAL(3),NEC_PIEDRA(3),NEC_MATERIAL(9),COMP_MADERA(0),COMP_PIEDRA(0
),COMP_METAL(0),COMP_ROBOT(0),COMP_BOMBAS(0),COMP_ANTI(0),COMP_VIGILANTE(0),COMP_TARJETA
S(0),COMPARACION(0)]
  ACCION: IR_POR_TARJETA(SOLDADO)
  EQUIPO_ACTIVIDADES: SOLDADO: YENDO_POR_TARJETA,BOMBER: ESPERANDO_ORDEN,ESPIA:
ESPERANDO_ORDEN
  EQUIPO_ENEMIGO_ACTIVIDADES: SOL_ENE: ESPERANDO_ORDEN,BOM_ENE:
ESPERANDO_ORDEN,ESP_ENE: ESPERANDO_ORDEN
  =====
  hay una tarjeta! y está de mi lado y muy cerca del límite del zona de peligro
  no imoprt a quier vaya por ella. Y el enemigo está lejos de mi.
  Quiero usar todo mi equipo para traer un solo material, creo que es mas rápido
  Mi estrategia inicial es acabar el juego bombardeando a la base enemiga y nunca
  atacar a los soldados obteniendo todo el material que represente el menor riesgo para mi
  equipo
  =====
  <
```

Fig. 4.1 Colección de datos guardados al hacer una tirada en el videojuego “Guns”.

En la anterior figura de información se distinguen dos partes esenciales:

1. La recopilación de la información contenida en el ambiente de juego en el momento de hacer un movimiento por parte de los usuarios (letras azules).
2. El registro de la acción tomada (letras verdes).
3. El comentario agregado por el usuario (letras rojas).

Una partida de juego está compuesta de varios de los anteriores registros almacenados en un solo archivo de texto.

4.2 Preprocesamiento de la experiencia previa

El preprocesamiento de toda la información obtenida en la fase de experimentación se hizo con un programa diseñado con ese propósito (ProcesaExperiencia.exe) y que fue explicado en el capítulo anterior. Debe decirse que aquí se entiende por preprocesamiento de la experiencia previa a la “traducción” de los comentarios agregados por los usuarios en un lenguaje común a una forma equivalente en significado, pero usando expresiones que puedan ser interpretadas y usadas por la implementación de la IA. Ahora bien, se dice que esta fase de procesamiento es un preprocesamiento pues deja lista la información para un segundo y último procesamiento de la misma.

Para el preprocesamiento se tuvo que analizar cada uno de los movimientos efectuados en cada partida a fin de encontrar una forma de expresar los comentarios incluidos por los usuarios mediante una estandarización de expresiones. Esto es, se examinó con cuidado cada una de las partidas almacenadas con el objetivo de investigar si el conjunto de alternativas y situaciones generadas por el ambiente dinámico del juego era cerrado y acotado.

De este análisis se encontró que, por muy sencillo que sea este ambiente dinámico, es muy pequeña la probabilidad de que se presenten dos situaciones o estados en el ambiente estrictamente iguales. Sin embargo, se descubrió que aun en estas circunstancias, la definición de ambiente de juego y las reglas que lo gobiernan, así como la naturaleza de su implementación hace que, en el caso de los jugadores, las “razones” por las cuales alguien tome una decisión pueden identificarse y enumerarse, puesto que se conoce “lo que puede suceder” y “lo que se puede hacer” dentro del ambiente de juego. Dicho de otra forma, las señales, eventos o factores que hacen reaccionar a quienes se desarrollan dentro de este ambiente es un conjunto finito y numerable (cerrado y acotado).

De manera que, al identificar y enumerar estos factores, se pudo ver que los comentarios agregados por los usuarios del porqué de sus decisiones al momento de estar jugando podían ser expresados mediante un conjunto de mnemonicos que al ser bien elegidos podrían ser interpretados y manejados por la implementación de la IA conservando su significado y esencia original. Esto es cierto en este caso, puesto que todos los estímulos (eventos y señales) percibidos en el juego por los jugadores (mediante los agentes que maneja) y que los hicieron reaccionar, están relacionados con

valores de variables del ambiente susceptibles de ser medidos gracias a su especificación al momento de la planeación del juego y sus reglas.

A la enumeración y especificación del correspondiente mnemonico de cada uno de los factores anteriormente mencionados e identificados se le conoce en este trabajo como la estandarización de la información, y representa su abstracción de un primer nivel.

Esto bien corresponde, desde el punto de vista del autor de este trabajo, a la abstracción que nosotros, las personas, hacemos de la información que recibimos de nuestro ambiente mediante una asignación de un concepto propio (palabra o idea clave) para su posterior uso e interpretación, completamente particulares de una persona en especial.

La estandarización de la información es contenida e implementada para su uso en el programa que ayuda a procesar esta información.

Específicamente, la estandarización comprende la definición de los cuatro elementos que componen la estrategia (modo de juego para los ataques a la base enemiga, a los soldados enemigos, atención a las tarjetas de poder y el uso del equipo), la causa principal por la que se tomó la decisión de dar una orden a un soldado en especial, lo que corresponde directamente a los comentarios agregados como texto por parte de los usuarios, así como el objetivo final (la forma de ganar el juego) que tenían en mente los jugadores y la actitud predominante del jugador al efectuar sus movimientos (defensivos u ofensivos). Tiene que mencionarse que en el caso particular de este ultimo aspecto, después de cada partida se les hizo preguntas específicas a los usuarios para poder definir su “actitud” al momento de jugar. De estas preguntas y sus respuestas se tomaron notas para poder definir el elemento “actitud” y poderlo agregar como parte del procesamiento de la experiencia previa.

De acuerdo con la especificación del modelo Ensimismamiento-Reflexión, a este cúmulo de archivos de texto de experiencia previa preprocesados se le conoce como la *representación concreta* del conocimiento.

En este punto debe hacerse la siguiente especificación: Como el modelo Ensimismamiento-Reflexión dicta, debe de existir una representación abstracta y una representación concreta de la información que funciona como experiencia previa. Sin embargo, la abstracción hecha como en este preprocesamiento, de ninguna manera corresponde a la representación abstracta del modelo, como se explicará en breve. Más bien, esta primera abstracción corresponde al ejercicio intelectual de las personas de

asignarle un lugar en la memoria a la información y de asignarle un concepto o “expresión clave” a información que tiene un significado e interpretación en si misma. Lo que es completamente diferente de la representación abstracta, la cual representa un cúmulo de información relacionada y acumulada en contextos o situaciones específicas propias de la naturaleza de la implementación del modelo.

Para ver el resultado de este preprocesamiento, el registro mostrado en la sección anterior se ve como sigue después de haber sido procesado con el programa ProcesaExperiencia:

```

>
  CONTEXTO_SOLDADO(Pos(40,35),Bomber(55,35): 15,Espia(70,35): 30,Sol_Ene(920,475):
984,Bom_Ene(935,475): 997,Esp_Ene(950,475): 1011,Tornado(No): ,Carta(373,263):
404,DCarta(13,287,63,87),Guardian(No): ,Guar_Ene(No): ,Robot(No): ,Robot_Ene(No):
,Trampa(10,10): 39,Trampa_Ene(1000,550): 1089,Bomba(No): ,Refugio(No): ,Ref_Ene(No):
,Mad_1(930,70): 891,Mad_2(100,500): 469,Met_1(460,280): 486,Met_2(520,240):
522,Pie_1(490,55): 450,Pie_2(510,415): 604,Base(20) Base_Ene(1000) TSole(Si,1,984)
TBomE(Si,1,997) TEspE(Si,1,1011) TTor(No,0,0) TCarta(Si,1,404) TBomba(No,0,0)
TGuarE(No,0,0) TTraE(Si,1,1089) TRefE(No,0,0) TMad1(1,891) TMad2(1,469) TMet1(1,486)
TMet2(1,522) TPiel(1,450) TPie2(1,604) xxxxxxxxxx TENSIONES(Vr: 1,Sr: 11)
xxxxxxxxxxx
  EDO_SOLDADO(Resist: 5,Nadie me ataca,,No ataco,,Trampa: ;Libre!,Guard:
;Libre!,YENDO_POR_TARJETA,Desarmado)
  BASE[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(43
),NEC_MADERA(3),NEC_METAL(3),NEC_PIEDRA(3),NEC_MATERIAL(9),COMP_MADERA(0),COMP_PIEDRA(0
),COMP_METAL(0),COMP_ROBOT(0),COMP_BOMBAS(0),COMP_ANTI(0),COMP_VIGILANTE(0),COMP_TARJETAS
(0),COMPARACION(0)] <...>
  BASE_ENEMIGA[Madera(0),Metal(0),Piedra(0),Bombas(0),Tarjetas(0),Antibombas(0),VR(4),SR(4
3),NEC_MADERA(3),NEC_METAL(3),NEC_PIEDRA(3),NEC_MATERIAL(9),COMP_MADERA(0),COMP_PIEDRA(0
),COMP_METAL(0),COMP_ROBOT(0),COMP_BOMBAS(0),COMP_ANTI(0),COMP_VIGILANTE(0),COMP_TARJETA
S(0),COMPARACION(0)]
  ACCION: IR_POR_TARJETA(SOLDADO)
  EQUIPO_ACTIVIDADES: SOLDADO: YENDO_POR_TARJETA,BOMBER: ESPERANDO_ORDEN,ESPIA:
ESPERANDO_ORDEN
  EQUIPO_ENEMIGO_ACTIVIDADES: SOL_ENE: ESPERANDO_ORDEN,BOM_ENE:
ESPERANDO_ORDEN,ESP_ENE: ESPERANDO_ORDEN
  =====
  hay una tarjeta! y está de mi lado y muy cerca del límite del zona de peligro
no imoprt a quier vaya por ella. Y el enemigo está lejos de mi.
  Quiero usar todo mi equipo para traer un solo material, creo que es mas rápido
  Mi estrategia inicial es acabar el juego bombardeando a la base enemiga y nunca
atacar a los soldados obteniendo todo el material que represente el menor riesgo para mi
equipo

  ESTRATEGIA: Continuos+Nunca+Oportunos+Total CAUSA: Distancia a tarjeta+Distancia
Tarjeta-Borde de zona de peligro+Distancia a equipo Rojo[] OBJ. FINAL: Bombardear base
ACTITUD: Ofensiva

  =====
  <

```

Fig. 4.2 Colección de datos capturados durante una tirada en el juego “Guns” procesada con el programa “ProcesaExperiencia”.

En la anterior tabla se muestra un registro de un movimiento hecho por un jugador ya preprocesada, y donde se han marcado en negritas la estandarización o abstracción de parte de la información contenida en esta, específicamente del comentario agregado como texto por parte del usuario.

Tal y como se puede apreciar, existe una correspondencia exacta entre el comentario hecho por el usuario y la información agregada con la ayuda del programa para el preprocesamiento. Por lo que se dice que la elección de la estandarización de la información es correcta.

4.3 Construcción de la representación abstracta del conocimiento

Como ya se ha explicado la representación abstracta del conocimiento (en el caso particular de este proyecto y por su adaptación) es el resultado de un segundo procesamiento de la información obtenida como experiencia previa. La representación abstracta representa una abstracción total del conocimiento previo tal que permite su simplificación e interpretación gracias a un orden específico de los datos y su disponibilidad (si aparecen registrados o no) y que es capaz de describir un estado o situación predominante en un momento definido dentro del ambiente de juego.

Específicamente hablando, la representación abstracta es la transformación de la información obtenida del ambiente y almacenada en la representación concreta a una forma simplificada de esta, capaz de describir un contexto o situación predominante en el ambiente de juego.

Esta traducción o transformación de la información consiste en extraer solo los datos relevantes de todo el cúmulo de datos obtenidos en un registro de una jugada, asociarlos a una palabra o expresión clave para su identificación de su significado, y ponerlos en un orden definido que ayuda a su manejo.

En otras palabras, siguiendo la especificación del modelo Ensimismamiento-Reflexión, la representación abstracta de la información en este trabajo es la traducción de la información contenida en un archivo de experiencia previa, filtrada por relevancia y expresada en términos de tensiones, eventos y causas relacionadas a la toma de una decisión por parte de un jugador en un momento específico.

Esta abstracción de la información se hace registro por registro de cada una de las tiradas hechas en una partida de juego, después de lo cual, la transformación resultante se guarda en estructuras llamadas átomos.

Un átomo no es más que una estructura de datos con formato definido que contiene la información abstraída correspondiente a una tirada en una partida de juego.

En general, cada átomo guardado en la representación abstracta, describe el contexto general predominante en el ambiente de juego al tomar una decisión el cual almacena entre otras cosas la causa de la decisión tomada por el jugador, la acción tomada y el estado actual del agente(s) involucrado(s) en la acción, así como el estado general del ambiente de juego.

Adicionalmente se tiene que mencionar que esta representación abstracta o segundo procesamiento de la información que representa la experiencia previa, difiere del preprocesamiento en el sentido en que, la primer transformación es una traducción de primer nivel, es decir, hace la traducción de una idea con significado propio a una palabra o concepto definido, y la segunda transformación es la reducción por selección o transformación de segundo nivel de toda la información contenida en un registro de una tirada del juego y capaz de describir una situación o estado completo. En otras palabras, el preprocesamiento tiene por objetivo la asignación de conceptos a ideas aisladas, y el procesamiento tiene por objetivo la descripción completa y detallada de situaciones o contextos de manera simplificada.

La construcción de la representación abstracta se efectúa accionando la opción implementada en el ambiente de juego para ello.

Después de haber dado la orden para iniciar la construcción de la representación abstracta y haber definido el archivo a procesar, se empieza la abstracción de la información obteniendo un átomo por registro contenido en el archivo que se está procesando.

El átomo correspondiente al registro mostrado en la sección anterior y preprocesado, sería el siguiente:

```
ATOMO #41
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ;Libre!,Guard:
Libre!,YENDO_POR_TARJETA,Desarmado)
Estrategia: Continuos+Nunca+Oportunos+Total
Causa: Carta(Si,1,404)DCarta(13,287,63,87)TSole(Si,1,984)TBomE(Si,1,997)TEspE(Si,1,1011)
Obj. Final: Bombardear base
Actitud: Ofensiva
Accion: IR_POR_TARJETA(SOLDADO)
```

Fig. 4.3 Ejemplo de la estructura átomo.

El proceso completo hasta este punto se puede resumir usando las tres anteriores tablas. Como se apuntó en la tabla 1, durante la fase de experimentación un usuario agregó un comentario cuando decidió enviar al agente “soldado” por la tarjeta que apareció en ese momento. En este comentario se especificó que se decidió de esa forma

porque el agente estaba “cerca” del lugar donde apareció la tarjeta, y porque esta estaba cerca de los límites de la zona de peligro, y porque por el estado actual del agente, se encontraba libre de peligro por la distancia entre él y el enemigo.

Durante la fase de preprocesamiento todo el anterior comentario fue estandarizado de la siguiente forma:

CAUSA: Distancia a tarjeta+Distancia Tarjeta-Borde de zona de peligro+Distancia a equipo Rojo[]

Fig 4.4 Estandarización de un comentario agregado por el usuario.

En donde se puede observar que esta estandarización corresponde exactamente al comentario agregado por el usuario, en donde cada uno de los aspectos que influyeron para tomar la decisión de enviar al agente soldado por la tarjeta de poder están separados por el signo “+”.

Durante la construcción de la representación abstracta se interpretaron estas frases con la ayuda de un archivo de estandarización. Esto es, que el modulo de abstracción abre un archivo en donde se define una correspondencia entre una frase de estandarización y un dato concreto capturado del ambiente de juego durante la tirada. De modo que el procedimiento general de abstracción aquí consiste en leer una frase que se relaciona con un dato guardado en un registro de jugada. Este dato se lee y se agrega a la estructura general llamada átomo. Lo que finalmente resulta en:

Causa: Carta(Si,1,404)DCarta(13,287,63,87)TSole(Si,1,984)TBome(Si,1,997)TEspE(Si,1,1011)

Fig. 4.5 Representación de un comentario agregado por el usuario en tiempo de juego en la estructura átomo.

La anterior tabla es parte del átomo y corresponde a la causa que influenció al usuario para la acción de ir por la tarjeta en ese momento dado. Las expresiones que componen la expresión contenida en la anterior tabla corresponde a

- 1) La existencia de la tarjeta, el valor de la tensión producida por la distancia entre el agente soldado y la tarjeta más la distancia expresada en pixeles.
- 2) la distancia de la tarjeta al límite superior, inferior, derecho e izquierdo de la zona de peligro expresada en pixeles.

- 3) la presencia del soldado enemigo, la tensión producida por la distancia entre este y el agente soldado y la distancia expresada en pixeles.
- 4) y 5) corresponden a los datos anteriores pero correspondientes a la relación con los agentes bombardero y espía enemigos.

Como se puede apreciar, la abstracción de los datos contenidos en lenguaje común y agregados por el usuario, han sido expresados (abstraídos) en valores numéricos con una interpretación definida y exclusiva de la implementación del modelo usado en este trabajo.

4.4 Cómo funciona la adaptación del modelo Ensimismamiento-Reflexión.

La intervención de la implementación de la adaptación del modelo Ensimismamiento Reflexión a este proyecto inicia cuando se acciona la opción “Usuario Vs. IA”.

Para explicar este, se mostrará en esta sección como trabaja la implementación del modelo con ejemplos específicos.

4.4.1 Generando la semilla inicial

El modelo debe iniciar creando una semilla inicial. Esta, como ya ha sido explicado puede ser creada de forma completamente aleatoria (donde cada elemento de la semilla inicial es seleccionado de entre todas las posibilidades aleatoriamente) o por selección aleatoria (se selecciona un átomo de la representación abstracta aleatoriamente de donde se extraen los datos que componen la semilla inicial).

La semilla inicial corresponde a la especificación de los cuatro elementos que definen la estrategia de juego (criterio de ataque a la base enemiga, a los agentes soldado, atención a las tarjetas de poder que aparezcan y el uso del equipo en cada tarea asignada) y el objetivo Final. Una vez especificada la semilla inicial, comienzan los ciclos entre los estados característicos del modelo adaptado.

En general, la forma en que interviene el modelo Ensimismamiento reflexión es el siguiente:

1. Se toma una “fotografía” del contexto general del ambiente de juego, es decir, se toman todos los datos que representan el estado actual del ambiente de juego.

2. Se construye una estructura asociativa, la cual, corresponde a la estructura de un átomo y que representa el contexto actual general del ambiente de juego.
3. Con esta estructura se empieza a comparar con los átomos almacenados en el archivo de la representación abstracta para poder encontrar correspondencias entre el contexto actual contenido en la estructura asociativa y alguno almacenado en este archivo de la representación abstracta para poder encontrar correspondencias. Si estas existen entonces se extrae la acción asociada al átomo que correspondió a la comparación.
4. Una vez obtenido el conjunto de posibles siguientes acciones se selecciona una de este conjunto. En este punto termina el estado de Ensimismamiento (incisos 2,3 y 4).
5. Finalmente, se valida la acción escogida. Esto es, se verifican las precondiciones y poscondiciones de la acción seleccionada, es decir, se verifica si existe en el ambiente de juego lo necesario para poder ejecutar la acción y se toman en cuenta las poscondiciones de la misma. Dichos datos (pre y poscondiciones se encuentra almacenadas en el archivo de “acciones primitivas” en donde se especifica el formato de cada acción, quien puede ejecutar dicha acción, así como las precondiciones y poscondiciones de la misma). Si las precondiciones no se cumplen (se lanza una excepción o advertencia) entonces se busca en el archivo de acciones primitivas una que asegure que las precondiciones (que en este proyecto han sido denominadas como “objetivos particulares”) se cumplan y si sí se cumplen las precondiciones entonces se agrega esta acción al conjunto de siguientes posibles acciones a tomar. Este proceso se repite hasta estar seguros de que la serie de acciones obtenidas no lance excepciones. A este último inciso se le llama el estado de Reflexión.

Siguiendo el anterior esquema, el paso inicial de una partida nueva es generar una semilla inicial automáticamente.

4.4.2 Determinando objetivos Particulares

Ahora bien, desde el punto de vista de la implementación de la IA, el siguiente objetivo lógico es generar un plan de juego. De modo que el modelo inicia con el estado de reflexión (un estado de validación y evaluación del material actualmente generado) tomando como referencia el objetivo final generado por la selección de la semilla

inicial. La disposición de los datos iniciales permite la verificación del plan de juego y de los objetivos particulares, representadas por estructuras de datos (estructuras inicialmente que en el punto inicial se encuentran vacías) lo que garantiza la generación de acciones como plan de juego apegándose a objetivos particulares definidos.

Como se puede ver, esta primera acción de Reflexión corresponde a lo que ordinariamente una persona hace al encontrarse en un sistema dinámico cambiante, que es evaluar el estado actual del ambiente para poder empezar a planear.

Debe decirse que el Objetivo final representa una acción de alto nivel, es decir, una acción que engloba un objetivo. Específicamente hablando, en este proyecto el objetivo final puede tomar dos valores: Bombardera Base o atacar soldados, y representa la forma en que un jugador puede ganar un juego. Ambas acciones pueden ser alcanzadas por medio de acciones particulares por parte de los integrantes del equipo, ya sea un bombardeo a la base por parte del agente bombardero o un ataque a algún soldado enemigo por parte de cualquier otro agente.

4.4.3 Creando el Plan de Juego

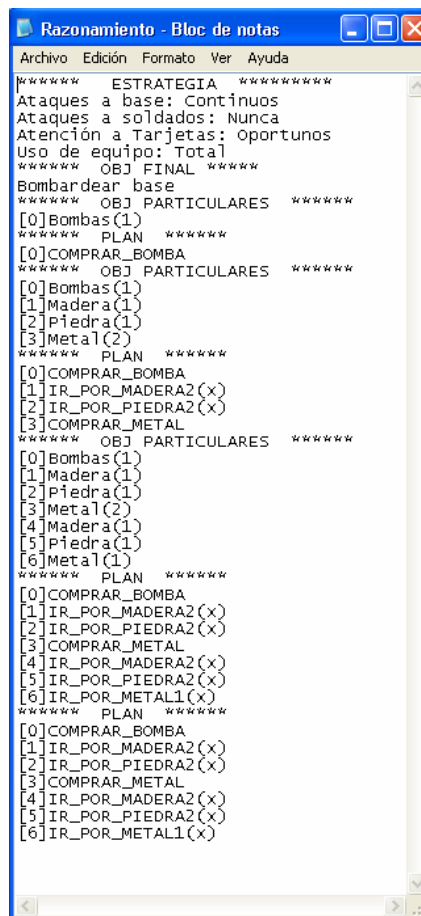
Tomando en cuenta lo anterior lo establecido en la sección anterior, el estado de reflexión inicial toma en cuenta el objetivo final y empieza por verificar si existen en el plan la acción que lleve a que se cumpla dicho objetivo así como las precondiciones necesarias para ejecutarla. Esto es, si por ejemplo, se tiene como objetivo final Bombardear la base enemiga, entonces se verifica si existe en el plan una acción que lleve a ese objetivo (en este caso BOMBARDEAR(Bomber). Si no es así, se agrega esta acción mediante entrar al estado de ensimismamiento y traer del archivo de acciones primitivas la acción que resuelva este problema. Si ya existe entonces se verifican las precondiciones, que en el caso de este ejemplo sería que hubiera almacenado en la base una bomba. Si estas precondición no existe entonces se agrega como objetivo particular.

Una vez mas entonces se hace la verificación de una acción que garantice el cumplimiento del objetivo particular y la validez de esta acción (si existen o no las precondiciones para poder llevarse a cabo).

Estos ciclos de verificación llevan a la formación de un plan un de una lista de objetivos particulares, la cual termina cuando todos los objetivos particulares pueden ser alcanzados con la lista de acciones contenidas en el plan de juego y que terminan con el cumplimiento del objetivo final.

Tiene que decirse además que la selección de acciones es fuertemente influida por la especificación de la estrategia de juego. Específicamente los dos primeros componentes de esta estrategia, que son el criterio de ataque a la base enemiga y a los soldados enemigos.

Puede hacerse un seguimiento de todo lo anteriormente explicado mediante un ejemplo. Cuando se elabora el plan de juego se forma un archivo de texto que tiene por objetivo hacer un seguimiento del “razonamiento” hecho por la implementación del modelo para formar el plan de juego. Un archivo de razonamiento ejemplo se ve como sigue:



```

Razonamiento - Bloc de notas
Archivo Edición Formato Ver Ayuda
***** ESTRATEGIA *****
Ataques a base: Continuos
Ataques a soldados: Nunca
Atención a Tarjetas: Oportunos
Uso de equipo: Total
***** OBJ FINAL *****
Bombardear base
***** OBJ PARTICULARES *****
[0]Bombas(1)
***** PLAN *****
[0]COMPRAR_BOMBA
***** OBJ PARTICULARES *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
***** PLAN *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA2(x)
[3]COMPRAR_METAL
***** OBJ PARTICULARES *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
[4]Madera(1)
[5]Piedra(1)
[6]Metal(1)
***** PLAN *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA2(x)
[3]COMPRAR_METAL
[4]IR_POR_MADERA2(x)
[5]IR_POR_PIEDRA2(x)
[6]IR_POR_METAL1(x)
***** PLAN *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA2(x)
[3]COMPRAR_METAL
[4]IR_POR_MADERA2(x)
[5]IR_POR_PIEDRA2(x)
[6]IR_POR_METAL1(x)

```

Fig. 4.6 Ejemplo de resultado del proceso que construye las estructuras “Objetivos_Particulares” y “Plan”.

Esta ilustración muestra paso a paso el resultado de la intervención el modelo Ensimismamiento Reflexión para poder definir tanto los objetivos particulares como el plan de juego.

Debe mencionarse que es necesario tener ambas estructuras (objetivos particulares y plan de juego) puesto que dentro de la implementación de este trabajo se

incluyó un módulo que constantemente verifica el cumplimiento de objetivos (particulares y/o finales). Si se cumple alguno de estos entonces se emite una señal para que el módulo de asignación de acciones entre en acción con la ayuda de las acciones cognitivas de tipo modelo, es decir, aquella acción evaluadora que elige al mejor candidato de entre los agentes disponibles para ejecutar la siguiente acción en el plan de juego. En seguida se explica cómo se generó la estructura que representa al conjunto de los objetivos particulares y el plan de juego mediante la intervención de la implementación del modelo Ensimismamiento-Reflexión. Para lo que la anterior ilustración se ha dividido en varios fragmentos.

4.4.4 Un ejemplo de Ejecución

4.4.4.1 Generación de la semilla inicial

Trabajando en este ejemplo específico con una carga aleatoria, se seleccionó aleatoriamente un átomo de la estructura abstracta de donde se extrajeron los 4 elementos de la estrategia de juego y el objetivo final. Esto se ve en la siguiente ilustración:

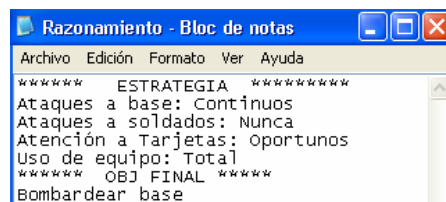


Fig. 4.7 Resultado de la selección aleatoria de la semilla inicial.

4.4.4.2 Iteraciones entre los estados Ensimismamiento-Reflexión.

Como ya se mencionó se inicia entrando al estado de reflexión para evaluar las condiciones actuales del ambiente de juego y crear un plan de juego influenciado por los objetivos particulares.

Puesto que el objetivo inicial es “Bombardera base”, el estado de reflexión “ve” si existe la tanto los recursos necesarios para completar el objetivo (1 bomba al menos) y una acción relacionado con este (Bombardear base). Puesto que inicialmente no existen estas condiciones el estado reflexivo entra en acción buscando en el archivo de acciones primitivas una acción que cumpla este objetivo. Una vez agregada se verifica si existen las condiciones para poder efectuarla. Una vez mas, puesto que se inicia con cero unidades almacenadas, entonces se entra una vez mas al archivo de acciones

primitivas y se agregan las precondiciones necesarias para ejecutarse la acción. Esto da por resultado:

```
***** ESTRATEGIA *****
Ataques a base: Continuos
Ataques a soldados: Nunca
Atención a Tarjetas: Oportunos
Uso de equipo: Total
***** OBJ FINAL *****
Bombardear base
***** OBJ PARTICULARES *****
[0]Bombas(1)
***** PLAN *****
[0]COMPRAR_BOMBA
***** OBJ PARTICULARES *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
```

Fig. 4.8 Resultado de la ejecución de un primer ciclo de la implementación de la adaptación del modelo.

Donde Comprar_Bomba satisface el objetivo final, y donde se agregan los objetivos particulares (precondiciones necesarias) para poder efectuar dicha acción. Llegado este punto se entra una vez mas al estado de reflexión.

Se evalúa entonces el estado de general del equipo, las precondiciones necesarias y el plan de juego. En este punto, no hay acciones que garanticen el cumplimiento de los objetivos particulares, por lo que se entra ciclos entre los estados ensimismamiento y reflexión para obtener las acciones necesarias y construir el plan adecuado para cumplir estos. El resultado es el siguiente:

```
***** ESTRATEGIA *****
Ataques a base: Continuos
Ataques a soldados: Nunca
Atención a Tarjetas: Oportunos
Uso de equipo: Total
***** OBJ FINAL *****
Bombardear base
***** OBJ PARTICULARES *****
[0]Bombas(1)
***** PLAN *****
[0]COMPRAR_BOMBA
***** OBJ PARTICULARES *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
***** PLAN *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA2(x)
[3]COMPRAR_METAL
```

Fig. 4.9 Resultado de la ejecución de un segundo ciclo de la implementación de la adaptación del modelo.

En este caso en particular se ve que hay una relación uno a uno entre los objetivos particulares y la acción que es candidata para cumplir con ese objetivo. Es de especial interés la acción [3] del plan de juego puesto que hace que se entre una vez mas en los ciclos entre los dos estados del modelo para poder satisfacer todas las condiciones para que el plan se considere completo.

Al entrar una vez más en acción la implementación del modelo se obtiene lo siguiente:

```

Razonamiento - Bloc de notas
Archivo Edición Formato Ver Ayuda
***** ESTRATEGIA *****
Ataques a base: Continuos
Ataques a soldados: Nunca
Atención a Tarjetas: Oportunos
Uso de equipo: Total
***** OBJ FINAL *****
Bombardear base
***** OBJ PARTICULARES *****
[0]Bombas(1)
***** PLAN *****
[0]COMPRAR_BOMBA
***** OBJ PARTICULARES *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
***** PLAN *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA2(x)
[3]COMPRAR_METAL
***** OBJ PARTICULARES *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
[4]Madera(1)
[5]Piedra(1)
[6]Metal(1)
***** PLAN *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA2(x)
[3]COMPRAR_METAL
[4]IR_POR_MADERA2(x)
[5]IR_POR_PIEDRA2(x)
[6]IR_POR_METAL1(x)

```

Fig. 4.10 Resultado de la iteración entre ciclos de la implementación de la adaptación del modelo.

En donde se ve que para comprar el metal se agregaron precondiciones como objetivos particulares y las acciones necesarias para poderlos cumplir.

La intervención consecutiva del modelo resulta pues en un plan completo que al ser verificado cumple todos los objetivos particulares y contiene el conjunto de acciones necesarias para poder alcanzarlos. Esto es:

```
***** ESTRATEGIA *****
Ataques a base: Continuos
Ataques a soldados: Nunca
Atención a Tarjetas: Oportunos
Uso de equipo: Total
***** OBJ FINAL *****
Bombardear base
***** OBJ PARTICULARES *****
[0] Bombas(1)
***** PLAN *****
[0] COMPRAR_BOMBA
***** OBJ PARTICULARES *****
[0] Bombas(1)
[1] Madera(1)
[2] Piedra(1)
[3] Metal(2)
***** PLAN *****
[0] COMPRAR_BOMBA
[1] IR_POR_MADERA2(x)
[2] IR_POR_PIEDRA2(x)
[3] COMPRAR_METAL
***** OBJ PARTICULARES *****
[0] Bombas(1)
[1] Madera(1)
[2] Piedra(1)
[3] Metal(2)
[4] Madera(1)
[5] Piedra(1)
[6] Metal(1)
***** PLAN *****
[0] COMPRAR_BOMBA
[1] IR_POR_MADERA2(x)
[2] IR_POR_PIEDRA2(x)
[3] COMPRAR_METAL
[4] IR_POR_MADERA2(x)
[5] IR_POR_PIEDRA2(x)
[6] IR_POR_METAL1(x)
***** PLAN *****
[0] COMPRAR_BOMBA
[1] IR_POR_MADERA2(x)
[2] IR_POR_PIEDRA2(x)
[3] COMPRAR_METAL
[4] IR_POR_MADERA2(x)
[5] IR_POR_PIEDRA2(x)
[6] IR_POR_METAL1(x)
```

Fig. 4.11 Resultado final de la iteración entre ciclos de la implementación de la adaptación del modelo.

Como acto final, se hace entonces una simplificación de ambas estructuras, tanto del plan de juego como de los objetivos particulares. Esto es, se suman por similitud los objetivos particulares (es decir se hace una suma total de un material requerido en particular y se expresa como un solo objetivo) y se simplifican las acciones borrando las que se repitan.

Esta acción de simplificación corresponde bien a la acción “normal” de una persona de agrupar conceptos y retenerlos en un “espacio” común.

4.4.4.3 Cómo intervienen las acciones cognitivas.

Una vez obtenido el plan completo y verificado las acciones cognitivas de tipo modelo entran en acción asignando a cada uno de los soldado las acciones contenidas en el Plan de juego. La acción cognitiva de asignación de tareas toma en cuenta el criterio de juego definido en los diferentes componentes de la estrategia de juego. Esto es, en el ejemplo que se esta siguiendo, se ve que la estrategia dice que “Nunca” se va a dirigir

ataques a los soldados enemigos, y que el uso del equipo es total. Esto quiere decir, que en posteriores intervenciones del modelo, nunca se van a incluir acciones como “ataca_soldado”. De la misma forma, todas las acciones susceptibles de ser ejecutadas por todos los soldados se van a asignar a todos ellos y van a ser ejecutadas por los que estén activos.

4.4.4.4 Los eventos en el ambiente de juego como banderas para la intervención de la implementación del modelo Ensimismamiento-Reflexión.

En el ambiente de juego suelen suceder eventos los cuales lanzan señales, mismas que son recibidas y reconocidas por los agentes que existen en ese momento en el juego. Dependiendo de la naturaleza de las señales y del agente, se puede iniciar la intervención de la implementación del modelo E-R. Los posibles eventos pueden ser tales como la aparición de una tarjeta de poder, la aparición de un tornado, la desaparición de un agente, entre otras.

Sin embargo, es esencial hacer una evaluación de la situación al momento de recibir una señal por parte de un agente. Esta evaluación con el propósito de ver y decidir si la estrategia original, los objetivos particulares definidos y el plan construido deben ser redefinidos.

Por ejemplo, cada vez que aleatoriamente (en tiempo y espacio) aparezca una tarjeta de poder se evaluará la estrategia definida, los Objetivos particulares así como la estructura que contiene el plan de juego. En el caso de que la estrategia con la que se esté jugando el elemento “atención a tarjetas” corresponda al criterio de “oportuno” entonces la evaluación se hace de la siguiente manera:

1. Se presenta el evento “aparece tarjeta”, lo que activa la intervención del modelo Ensimismamiento Reflexión.
2. Se toma una “fotografía” del ambiente actual en el juego y se construye una estructura asociativa.
3. Puesto que fue el evento relacionado con la tarjeta la que activó la intervención del modelo, la estructura asociativa es filtrada (modificada) de manera que quede como criterio de comparación los datos relacionados con la tarjeta.
4. Con esta estructura se entra al estado de ensimismamiento y se buscan un átomo que corresponda a las condiciones y criterios actuales. De este modo, lo que alguna vez alguien consideró oportuno (si es que existe el registro

correspondiente) al ver una tarjeta, puede usarse como el criterio de “oportuno” para la toma de una decisión en este ejemplo.

5. De todas las coincidencias obtenidas en el paso (4) se selecciona una para que pueda ser la siguiente acción posible. Esta selección puede ser aleatoria o puede ser filtrada. La selección aleatoria es simplemente, como su nombre lo indica, seleccionar una acción al azar del conjunto obtenido en el paso (4). Filtrar quiere decir quitar todas aquellas que no puedan ser tomadas instantáneamente en ese preciso momento lo que requiere de la intervención de la evaluación hecha en el estado de Reflexión. En este punto se tiene entonces que mencionar que algunas de las acciones tomadas por la acción del modelo deben ser reactivas, pues no hay tiempo, por el dinamismo del ambiente, de asegurar el cumplimiento de las precondiciones necesarias para llevar a cabo una acción, a semejanza de lo que hace una persona al encontrarse con situaciones de estrechez y tensión, cuando hay que tomar una decisión aunque no sea la ideal por las condiciones existentes.

Los anteriores 5 pasos aplican también a eventos tales como “aparición de tornado”, “soldado atrapado” (un soldado ha caído en una trampa) y “soldado detenido” (un soldado ha sido detenido por un guardián). En todos estos casos, se entra al estado de Ensimismamiento con la estructura asociativa modificada de acuerdo al evento para recuperar todas las posibles siguientes acciones y se toma la decisión entrando ayudado por el estado de reflexión.

Ahora bien, en este caso en particular se toma en cuenta el siguiente hecho:

Pudiera ser, que dado un evento, este active la intervención del modelo Ensimismamiento Reflexión para saber que hacer, y puede ser que la decisión tomada no vaya de acuerdo por el plan de juego establecido, de manera que, para garantizar que los criterios de juego sigan siendo respetados (la estrategia de juego siga siendo el criterio de juego) y para que el plan no sea modificado, se hizo y siguió el siguiente razonamiento.

En un ambiente dinámico, cuando se interacciona en el, y específicamente cuando se tiene un objetivo y un plan de acción, por la misma naturaleza del ambiente cabe la posibilidad de la aparición de imprevistos (eventos que afecten el plan original) tales que no se pueden ignorar por la importancia que representan en si mismos, es decir que pueden poner en peligro el éxito del plan y que no se pueden evitar pues forman

parte del ambiente como tal. En tal caso, la mejor de las opciones sería tomar acciones alternativas que ayuden a librar el “problema” sin modificar el plan, para lo que se hace una memoria a corto plazo. Es decir, dado un evento que no se pudo pasar por alto, debe ser superado con una acción alternativa en el momento, pero, se debe de “tener en mente” el plan original, para que una vez arreglado el problema se retome la el plan original.

Para esto se implementó en cada uno de los agentes, una memoria a corto plazo ayudado por una acción cognitiva de tipo modelo. Estas dos características funcionan de la siguiente forma:

1. Se tiene un plan establecido y un objetivo final.
2. Se da un evento que pone el peligro el seguimiento del plan.
3. Entra en acción la implementación del modelo para poder encontrar la acción que ayude a resolver el problema.
4. Si se tiene éxito en el paso (3) entonces, se abandona la acción actual, es decir, se evalúa si esta acción sigue siendo necesaria para alcanzar los objetivos particulares, si es así, entonces se agrega una vez más al plan de juego y se asigna la acción encontrada en el paso (3) y se ejecuta.
5. Una vez ejecutada la acción y superado el problema, se retoma la asignación de tareas contenidas en el plan de juego siguiendo los criterios de la estrategia definida.

Todas estas características han sido implementadas con éxito y han sido probadas una vez mas en una segunda fase de experimentación con algunos de los voluntarios que estuvieron en la primera fase de experimentación.

Los resultados has sido satisfactorios. En contadas ocasiones la IA ha ganado partidas y ha tomado decisiones inesperadas, como el desplazamiento de los agentes a lugares específicos dentro del tablero o ha tomado decisiones muy arriesgadas que le cuesta la “vida” a alguno de los agentes que controla.

Resumen

En este capítulo se presentó de forma detallada y mediante ejemplos de ejecución como la implementación descrita y planeada en el capítulo anterior trabaja, los resultados obtenidos en cada una de sus fases y el trabajo hecho para hacer posible la implementación de este trabajo.

Así mismo se presentaron los algoritmos seguidos para la implementación del modelo, las semejanzas y diferencias con el modelo original.

Capítulo 5

Resultados y Evaluación

En este capítulo se presenta la fase de evaluación mediante experimentación de la implementación de la adaptación del modelo Ensimismamiento-Reflexión a la generación de estrategias de juego para el juego de computadora descrito en el capítulo anterior.

5.1 Evaluación

El proceso de evaluación consistió en experimentación directa. Esto es, una vez finalizada toda la implementación descrita en el capítulo anterior, se pidió la ayuda de voluntarios para que pudieran jugar una partida con el juego desarrollado en esta investigación.

Esta segunda fase de experimentación de esta investigación difiere de la primera en el hecho de que inicialmente, quienes se enfrentaban en una partida eran dos personas, pero en esta segunda fase experimental, se enfrentan una persona y la implementación de la adaptación del modelo Ensimismamiento-Reflexión.

La evaluación consiste en observar el desempeño del modelo desde dos diferentes perspectivas:

- 1) Desde el punto de vista de un espectador, quien observara el desarrollo de una partida.
- 2) Desde el punto de vista de quien entendiera la implementación hecha del modelo, analizando así mismo la captura de datos hecha durante esta segunda fase de experimentación.

Ambos puntos de vista se conjugan para hacer los comentarios finales al mismo tiempo que se presentan los resultados obtenidos.

5.2 Resultados

Es de mucho interés el observar la manera de “reaccionar” ante los eventos por parte de la IA implementada. La percepción de las señales emitidas por eventos como los son la *Aparición de un Tornado*, *Aparición de una tarjeta de Poder*, y *Relocalización de la trampa enemiga*, pudieron ser detectados en el instante debido y manejados la mayoría de las veces exitosamente. Con esto se quiere decir, que observando como espectador, la reacción de los agentes controlados por la IA implementada es muy semejante a la que tuviera un jugador humano debido a la evaluación de la situación y la elección de una acción.

En todas las ocasiones los agentes evitaron enfrentarse a un peligro inminente esquivando los obstáculos.

Desde el punto de vista del desarrollador de la implementación es de gran interés el hecho de que en las ocasiones cuando se requirió reaccionar inmediatamente ante un evento que amenazaba la seguridad de alguno de los agentes del equipo controlado por la IA, la acción tomada permitió conservar la estrategia escogida originalmente y fue seguida hasta la culminación de la partida. Esto es, que aunque la reacción de los agentes pudiera afectar el plan trazado para seguir la estrategia, esto solo ocurrió en una de las 7 partidas celebradas en esta fase de evaluación.

Sin embargo, en las dos ocasiones en que la reacción elegida por la implementación del modelo para cada uno de los agentes que controlaba difirió del plan de juego para seguir la estrategia original, el mismo modelo, es decir, la iteración entre sus dos estados característicos y el proceso de constante evaluación de la Estructura Maestra, permitió una redefinición de toda la información contenida en esta estructura de tal forma de la estrategia y el plan fue redefinido para tratar de alcanzar la meta final, ganar la partida. Todo esto fue ejecutado de manera satisfactoria.

De lo todo lo anteriormente mencionado, en seguida se documentan algunas de las pruebas hechas y que se consideran representativas para esta sección.

5.2.1 Elección de la Semilla Inicial:

La selección de la semilla inicial se hizo como una Selección Aleatoria sobre Representación Abstracta, lo que quiere decir, que esta se seleccionó generando un número aleatorio entre 0 y el total de átomos contenidos en la Representación Abstracta (RA). El número resultante representa entonces el número de átomo en la RA del que se toman los parámetros necesarios para definir una estrategia de juego.

El número aleatorio es igual a 0. El átomo 0 de la RA, es el siguiente:

```
ATOMO #0
ESTADO: {Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_TARJETA,Desarmado}
Estrategia: Continuos+Nunca+Oportunos+Total
Causa: TCarta(Si,1,404)DCarta(13,287,63,87)TSolE(Si,1,984)TBomE(Si,1,997)TEspE(Si,1,1011)
Obj. Final: Bombardear base
Actitud: Ofensiva
Accion: IR_POR_TARJETA(SOLDADO)|
```

Fig. 5.1 Átomo seleccionado aleatoriamente para la especificación de la semilla inicial

Los datos extraídos son los 4 componentes que conforman propiamente el criterio de juego “Estrategia”, El objetivo final y la Actitud de juego.

La extracción de los datos queda como sigue:

Estrategia	
Ataque Base enemiga	Continuos
Ataque soldados enemigos	Nunca
Atención tarjetas de poder	Oportunos
Uso del equipo	Total
Objetivo final	
Objetivo Final	Bombardera Base
Actitud	
Actitud	Ofensiva

Fig. 5.2 Resumen de los parámetros que definen la Estrategia de juego.

5.2.2 Intervención de la implementación del la Adaptación del Modelo E-R para la generación del Plan de Juego

El objetivo final seleccionado es Bombardera la Base, lo que indica que la forma en que se quiere ganar la partida es así, mediante bombardeos a la base enemiga, por lo que se tienen que armar un Plan que vaya de acuerdo con este objetivo.

El procedimiento que ahora tiene lugar es el descrito con detalle en la sección 4.4.4.2. Para formar el plan de juego. El resultado se puede ver en la siguiente ilustración.

```

*****  ESTRATEGIA  *****
Ataques a base: Continuos
Ataques a soldados: Nunca
Atención a Tarjetas: Oportunos
Uso de equipo: Total
*****  OBJ FINAL  *****
Bombardear base
*****  OBJ PARTICULARES  *****
[0]Bombas(1)
*****  PLAN  *****
[0]COMPRAR_BOMBA
*****  OBJ PARTICULARES  *****
[0]Bombas(1)
[1]Madera(1)
[2]Piedra(1)
[3]Metal(2)
*****  PLAN  *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA1(x)
[3]IR_POR_METAL2(x)
*****  PLAN  *****
[0]COMPRAR_BOMBA
[1]IR_POR_MADERA2(x)
[2]IR_POR_PIEDRA1(x)
[3]IR_POR_METAL2(x)

```

Fig. 5.3 Resultado de la generación del Plan de Juego

Como se puede apreciar en la Figura anterior la iteración entre los dos estados característicos del modelo E-R da por resultado un plan coherente para con los parámetros que definen los criterios de juego.

5.2.3 Asignación de Acciones

Las acciones cognitivas evalúan los criterios de juego y el estado general en el que se encuentran los tres agentes que componen inicialmente el equipo controlado por la IA. De esta evaluación se obtiene que:

- 1) El uso del equipo es Total, lo que significa que en lo posible se va a procurar mantener el equipo unido haciendo la misma actividad, que nunca se van a atacar a los soldados del equipo contrario y que si se aparecen tarjetas de poder, estas se van a atender solo en el caso de que sea oportuno hacerlo.
- 2) Que todos los miembros del equipo están esperando recibir una orden.
- 3) Que las actividades, a excepción de la actividad de “comprar Bomba” pueden ser ejecutadas por todos los agentes.

Tomando en cuenta la anterior evaluación, la sección de asignación de acciones en la Estructura Maestra queda como sigue:

Asignación de Acciones		
Soldado	Bombardero	Espía
IR_POR_METAL2	IR_POR_METAL2	IR_POR_METAL2

Fig. 5.4 Resultado de la asignación de acciones

Acto seguido, los agentes comenzaron a acumular metal, trayéndolo del punto de abastecimiento 2.

En este punto, al encontrarse dentro de la Estructura Maestra el Objetivo particular de Comprar una Bomba, las acciones cognitivas de constante evaluación “saben” cual es la cantidad de material requerido. Así que cada vez que haya un cambio en el material almacenado en la Base de control se evaluará si es suficiente el material que actualmente se está acumulando.

De modo que los agentes comienzan a moverse “concientes” de sus metas a corto y largo plazo, dotados de una constante actividad de evaluación del ambiente a

nivel personal y una acción cognitiva evaluadora general conciente del estado completo del ambiente.

Toda la anterior descripción se ve en pantalla como se muestra en la siguiente ilustración, en donde se puede apreciar a los tres agentes moviéndose juntos hacia el punto de abastecimiento de Metal número 2.



Fig. 5.5 Equipo controlado por la IA ejecutando la acción IR_POR_METAL2

Cada vez que se cumple una meta específica como la de tener el material almacenado requerido, todo el anterior proceso de evaluación para la asignación de acciones tiene lugar. Se borra la acción ya ejecutada del plan de juego y se reasigna la nueva acción en el plan de juego.

Las acciones cognitivas de evaluación general nunca dejan de ejecutarse.

La asignación de tareas siempre se ejecutó con éxito.

5.2.4 Reacción a los eventos en el ambiente de juego

En un momento durante la ejecución de la acción de Ir por Metal al punto de abasto numero 2, aparecen dos eventos:

1. Aparece tornado.
2. Aparece Tarjeta de poder.

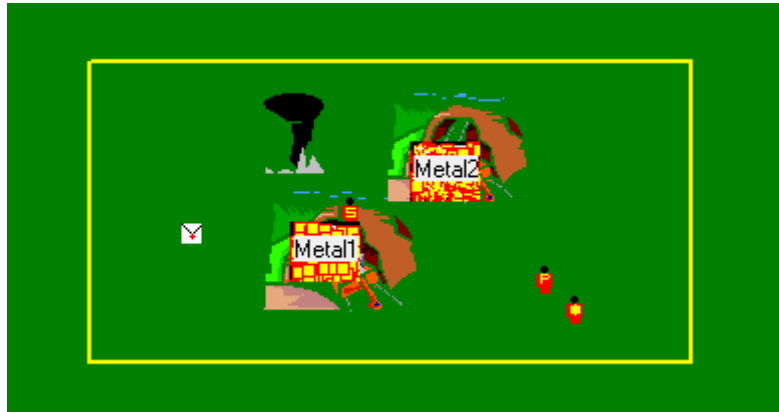


Fig. 5.6 Lanzamiento de los eventos “Tornado” y “Tarjeta”

En este caso particular se presenta primero el evento “Tornado”, seguido del evento “Tarjeta”.

En el momento en que tiene lugar cada evento se crea actualiza el contexto de cada uno de los jugadores y se crea una Estructura Asociativa por cada uno de ellos. El resultado es el siguiente:

EA de agente soldado

```

CONTEXTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,0,3,4)
CONTEXTO_P: [NO,3,NO,NO,840,320,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Total

```

EA del agente Bombardero

```

CONTEXTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,0,3,4)
CONTEXTO_P: [NO,3,NO,NO,425,1173,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Parcial

```

EA del agente Espía

```

CONTEXTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,0,3,4)
CONTEXTO_P: [NO,3,NO,NO,425,1173,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Parcial

```

Fig. 5.7 Estructuras Asociativas de los tres agentes al momento de lanzarse el evento “Tornado”

5.2.4.1 Procesando el evento “Aparece tornado”

Como se ha hecho notar e el capítulo 4 de este trabajo, cuando tiene lugar un evento, en este caso particular el evento “Tornado” se filtra le información contenida en cada una de las Estructuras asociativas de tal forma que contengan la información

necesaria para hacer la comparación entre la estructura asociativa resultante de la transformación debida al filtro que se ha aplicado y la Representación Abstracta de la Información.

La primera transformación de las estructuras asociativas presentadas en la sección anterior es la misma para las tres, es decir, se agrega el filtro del evento tornado, y queda como se ve en la siguiente ilustración:

```

CONTEXTTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,0,3,4)
CONTEXTTO_P: [NO,3,NO,NO,840,320,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Total
Filtro: TTor

```

Fig. 5.8 Primear transformación para una EA para el evento Tornado

Como se puede apreciar en la ilustración anterior, un renglón (que se ha escrito en color rojo) al final de la Estructura Asociativa se ha agregado para indicar el tipo de filtro que se aplicará a esta estructura Asociativa. Además se ha resaltado el segundo dato en el renglón del contexto Particular (*CONTEXTTO_P*) de los agentes la tensión debida a la aparición de un tornado, cuyo valor es 3. Este valor es debido a la cercanía del tornado a cada uno de los agentes del equipo.

Ahora es tiempo de que los estados E-R empiecen a iterar para poder tomar una decisión en cuanto a la situación presente. Pero antes de entrar en esta iteraciones, cada uno de los agentes ejecuta una de las acciones cognitivas a nivel personal, la cual tiene por objeto evaluar el estado personal de cada agente. Esta acción tiene por propósito determinar si el evento sucedido en el ambiente es del tipo que ponen en riesgo la seguridad de cada agente. En este caso particular, el evento tornado, si lo es, por lo que se pasa a la siguiente acción cognitiva de evaluación.

La siguiente acción cognitiva evalúa si algún agente de peligro está en su trayectoria. Esto es, el agente, personalmente evalúa si el tornado recién aparecido se encuentra en una posición tal que la trayectoria tomada por el agente para realizar su actual actividad puede cruzarse con la posición actual del tornado.

En este caso, como la Ilustración [tal.tal] lo indica, la posición actual del Tornado no representa un riesgo para ninguno de los agentes.

Esta evaluación hace que la intervención de la Implementación del modelo E-R sea detenida y no entre en acción pues no tiene razón por la cual ejecutarse.

5.2.4.2 Procesando el evento “Aparece Tarjeta”

Todo evento comienza por generar una Estructura Asociativa (EA) por cada uno de los agentes existentes (vivos). Las EA’s obtenidas para el evento “tarjeta” después de habersele agregado el filtro de este evento son las siguientes:

EA del agente Soldado

CONTEXTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,3,4)
CONTEXTO_P: [(100,210,40,30),1,NO,NO,451,1152,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Parcial

TCarta

EA del agente Bombardero

CONTEXTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,3,4)
CONTEXTO_P: [(100,210,40,30),1,NO,NO,225,149,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Parcial

TCarta

EA del agente Espía

CONTEXTO_G: BASE(12,1,2,0,1,0,4,49,0,3,3,6,0,2,2,0,0,0,0,2)
BASE_ENEMIGA(10,0,5,0,0,0,4,46,1,3,2,6,1,0,0,0,0,0,3,4)
CONTEXTO_P: [(100,210,40,30),1,NO,NO,721,582,1,1,1,NO,NO,NO,1,NO,2,1,1,1,1,1,12]
ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_METAL2,Desarmado)
Estrategia: Oportunos+Nunca+Oportunos+Parcial

TCarta

Fig. 5.9 EA para los tres agentes del equipo controlado por la IA para el evento “Tarjeta”

La aparición de todo evento acciona la evaluación personal de cada agente en este momento, y el resultado de esta evaluación es que no existe peligro por trayectoria para cada agente en el ambiente de juego. Sin embargo, el tipo de evento, en este caso la aparición de una tarjeta da paso a la intervención del modelo del proceso creativo, así que se empiezan las iteraciones entre los dos estados característicos del modelo E-R.

5.2.5 Ensimismamiento

El estado Ensimismamiento comienza por hacer comparaciones entre las Estructuras Asociativas obtenidas y la Representación Abstracta (RA) de la información que se tiene por Experiencia Previa. Esta comparación es guiada por el filtro que se agregó en la sección anterior. Es decir, la primera comparación que se hace es para encontrar todos aquellos átomos contenidos en la Representación Abstracta, tales que

tengan en su sección de “Causa” el filtro que se acaba de agregar a las EA recién creadas.

Esta primera comparación arroja como resultado el siguiente átomo contenido en la R-A, que es la única correspondencia a la comparación hecha:

ATOMO #12
 ESTADO: (Resist: 5,Nadie me ataca,,No ataco,,Trampa: ¡Libre!,Guard: ¡Libre!,YENDO_POR_PIEDRA1,Desarmado)
 Estrategia: Oportunos+Nunca+Oportunos+Parcial
Causa: TCarta(Si,1,452)DCarta(120,180,120,30)
 Obj. Final: Acumular
 Actitud: Defensiva
Accion: IR_POR_TARJETA(BOMBER_ENEMIGO)

Fig. 5.10 Resultado obtenido de una primera búsqueda durante el estado de Ensimismamiento.

Después del proceso de filtrado, que se explicó a detalle en el capítulo 3 de este trabajo (Ver sección 3.11.1 y 3.11.2) una estructura Asociativa es la siguiente:

FILTRO: TCarta(Si,1,322)DCarta(**100,210,40,30**)
 Estrategia: Oportunos+Nunca+Oportunos+Parcial
TCarta

Fig. 5.11 Resultado del filtro de Información en una EA debida al evento “Tarjeta”

Al hacer la comparación descrita también ya con detalle en la sección 3.11, el resultado es:

Posibles siguientes acciones	
Indice	Acciones
[1]	IR_POR_TARJETA(x)

Fig. 5.12 Conjunto de posibles acciones siguiente obtenidas del estado de Ensimismamiento

5.2.6 Reflexión

El estado de Reflexión valida los prerequisites de la acción seleccionada como la siguiente a ejecutar, que en este caso es una sola, y decide que no existe problema alguno para decidir que esta sea la siguiente acción a ejecutar.

5.2.7 Asignación de Acciones

La acción cognitiva que hace la asignación de acciones determina que, los tres agentes en este estado se encuentran en la misma rango de riesgo por la proximidad de los agentes que los rodean debido a la posición espacial que ocupan.

Evalúa además que son solo es el agente soldado y Bombardero los que pueden recoger las tarjetas, así que aleatoriamente elige entre estos dos y la sección asignación de acciones en la Estructura Maestra queda como sigue:

Asignación de tareas		
Soldado	Bombardero	Espía
Ir_Por_Trajeta(Soldado)	Ir_Por_Metal2(Bomber)	Ir_Por_Metal2(Espía)

Fig. 5.13 Estructura de asignación de tareas resultante después del estado Reflexión

Una vez detectado el cambio en la sección de asignación de tareas, se le da la orden al soldado de ir por la tarjeta.

Todos los eventos son manejados de la misma manera, y habiendo hecho las pruebas debidas se concluye que funciona correctamente la implementación de la adaptación del modelo Ensimismamiento-Reflexión hecha en este trabajo.

Resumen

En este capítulo se presentaron los resultados obtenidos en experimentación real de la implantación final de la adaptación del modelo Ensimismamiento Reflexión mediante ejemplos representativos de las pruebas hechas.

Así mismo se consideró la fase de prueba como parte de la evaluación del trabajo realizado en este proyecto de investigación, terminando en una conclusión satisfactoria con respecto al trabajo realizado.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones obtenidas en este trabajo, en la sección () se responden las preguntas planteadas en la introducción de este trabajo. Luego, en la sección () se revisan algunas características de la adaptación del modelo Ensimismamiento-Reflexión hechas en este trabajo y que se consideran aportaciones al mismo y representan algunos aspectos a considerar en trabajos futuros.

6.1 Respuestas a las preguntas introductorias

A lo largo de este trabajo se ha explicado cómo fue implementado el modelo del proceso creativo de Pérez Y Pérez para aplicarlo en la generación de estrategias de juego en un videojuego de computadora. Además se ha visto cómo funciona esta la adaptación del modelo y en el capítulo anterior se han presentado ya algunas conclusiones desde el punto de vista de una serie de evaluaciones realizadas al programa.

Esta sección está enfocada a contestar preguntas de investigación planteadas en la introducción de este trabajo. Las respuestas recogen todas las observaciones realizadas durante las sucesivas iteraciones de análisis, implementación y evaluación realizadas en el desarrollo de la investigación.

6.1.1 ¿Cómo representar el conocimiento?

Este es una pregunta que tuvo que ser resuelta en el análisis de la investigación, pues era necesario determinar las estructuras adecuadas para manipular la información usada por el sistema.

Por la naturaleza dinámica de este proyecto, no solo se tuvo que definir la forma más conveniente para el manejo de la información, sino también una forma de percepción de la misma. Lo que quiere decir que, puesto que en el ambiente de juego ocurren eventos los cuales activan a la implementación de la adaptación del modelo (la IA), se tuvo que implementar la forma de percibir e identificar dichas señales emitidas por el ambiente, lo cual también corresponde al establecimiento de la forma del manejo de la información.

Por lo anteriormente mencionado, en este trabajo fue necesario encontrar un esquema general para los “agentes” que se existiría en el ambiente de juego tomando en cuenta su tipo y su rol. A partir de un esquema, se hizo una diferencia entre los agentes que serían controlados por los jugadores y los que residirían como parte de la implementación del ambiente y que no sería controlado por algún usuario. Se estableció entonces la forma de interacción con el ambiente y la forma de relacionarse unos para con los otros, así como las acciones que se podían realizar y lo que podía suceder dentro del escenario de juego.

El esquema general de un agente que existe en el ambiente de juego y su interacción con este es el siguiente:

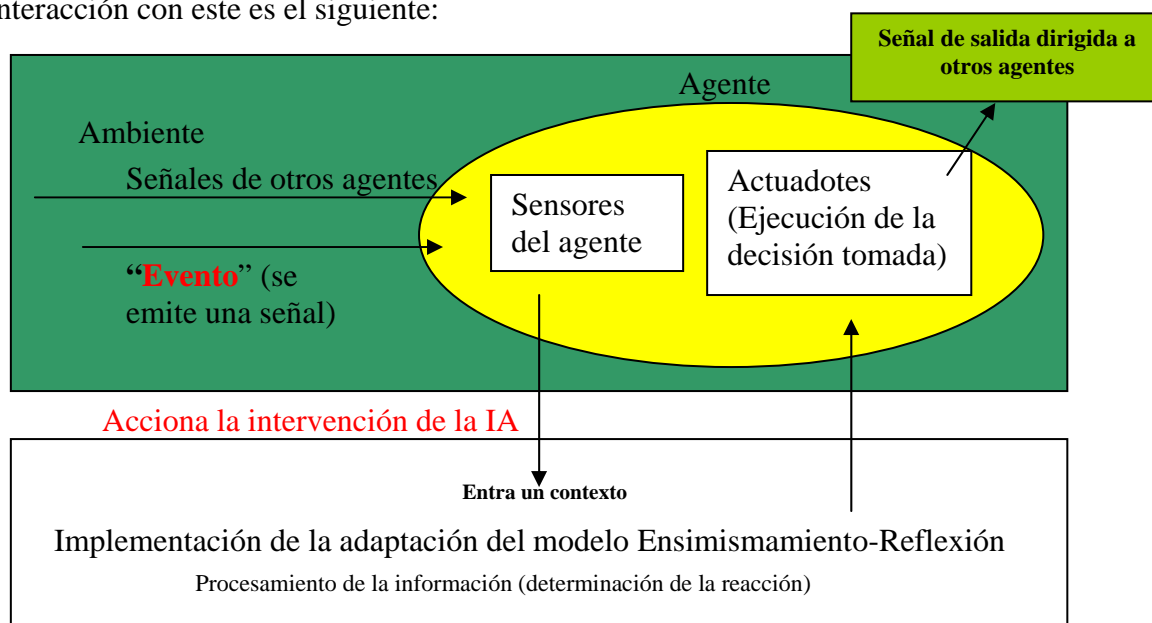


Fig. 6.1 Esquema general de un agente implementado en este proyecto de investigación.

Como se puede notar en la anterior ilustración, un agente debe ser dotado de las “habilidades” de percepción y de reacción. Estas habilidades deben de existir en dos sentidos: 1) relación ambiente-agente y 2) relación IA-agente. En el primer sentido se estableció una coordinación entre las señales emitidas por cada evento dentro del ambiente y la percepción de estas por parte del agente. En el segundo sentido se especificó la forma de comunicación entre la implementación del modelo usado y el agente.

Para la comunicación entre la implementación de la adaptación del modelo Ensimismamiento-Reflexión y el agente se usó la estructura “Contexto” y que esta compuesta de la colección de todos los datos trascendentes arrojados por el ambiente para poder tomar una decisión.

La estructura Contexto es usada para representar situaciones o estados en el ambiente dinámico de juego. La estructura contexto tiene por propósito contener información, parcial o completa, sobre la situación actual. Por lo tanto se usa una de estas estructuras por evento ocurrido en el ambiente y como estructura modular de comparación para buscar en memoria situaciones similares a las que se busca resolver.

6.1.1.1 Conclusión 1: La estructura contexto resultó eficaz y apropiada para la representación de estados en el juego.

Para el manejo de la información que finalmente controla al equipo que depende de la IA, se implementó una estructura dinámica (que se dimensiona, transforma y actualiza en tiempo de juego) llamada “Estructura Maestra” en la cual residen los criterios de juego (estrategia) y el plan de juego, y es una estructura fija en el sentido que representa las decisiones definitivas y finales hechas por el modelo que controla la parte inteligente.

6.1.1.2 Conclusión 2: En el caso particular de este proyecto, donde el ambiente es dinámico, la Estructura Maestra, que guardaba el modo y criterios de juego fue eficaz.

6.1.2 ¿Es posible la adaptación del modelo creativo Ensimismamiento-Reflexión que originalmente fue implementado en un contexto literario a un videojuego de computadora?

Por los resultados obtenidos, la respuesta a esta pregunta original de investigación es completamente afirmativa, con la certeza de que pudo ser adaptado el modelo creativo Ensimismamiento-Reflexión a este ambiente dinámico como el de un videojuego.

¿Es capaz de enfrentar la adaptación del modelo Ensimismamiento-Reflexión situaciones desconocidas para ella?

Como se pudo ver en la sección de evaluación de este trabajo, la probabilidad de que se puedan tener dos veces una misma situación como para tomar la misma decisión es casi nula. Por lo que se dice que vez tras vez las situaciones a las que se tuvo que enfrentar la IA siempre fue nueva.

Como se vio en la sección de evaluación, la adaptación de la implementación del modelo Ensimismamiento-Reflexión no solo pudo reconocer las circunstancias en las que se encontraba, sino proponer cada vez una solución a esta. De esto se concluye que:

La adaptación hecha aquí del modelo Ensimismamiento-Reflexión es capaz de proponer soluciones adecuadas a situaciones desconocidas.

6.2 Aportaciones de esta investigación

Esta investigación tiene como punto de partida y base teórica el modelo del proceso creativo Ensimismamiento-Reflexión aplicado originalmente a un contexto literario por Rafael Pérez y Pérez, el mismo que ha sido adaptado a la generación de estrategias de juego para un videojuego de computadora, por lo que evidentemente existen diferencias fundamentales. Algunas de las más significativas son las siguientes:

- Originalmente, el modelo Ensimismamiento-Reflexión (E-R) empieza a trabajar con una semilla inicial la cual es proporcionada por el usuario. En este trabajo, la semilla inicial es la situación inicial de juego.

- La interacción de E-R está guiada, entre otros parámetros, por las tensiones y relaciones. En este trabajo solo se hace uso de tensiones.
- La estructura general contexto en E-R está expresada en términos de personajes existentes, tensiones y relaciones. En este trabajo un contexto está expresado en términos de agentes existentes y tensiones.
- Del mismo modo, la adaptación de E-R conserva los aspectos fundamentales del modelo, entre los que destacan:
 - El uso de una estructura contexto para la representación de la situación predominante en un momento determinado.
 - Enfrentarse a circunstancias no contenidas en su experiencia previa.
 - El agrupamiento de acciones en una sola acción (plan de juego).
- A continuación se revisan las aportaciones al modelo E-R.

6.2.1 Aportaciones

Se considera una aportación la comprobación de la factibilidad del modelo E-R funcionando dentro de ambientes dinámicos como lo es el videojuego hecho y usado en esta investigación. Esto gracias a la implementación de un proceso de constante evaluación guiado por estímulos provenientes del medio ambiente.

La aportación directa que se deriva del anterior comentario es debida a la división de dos niveles en los cuales puede trabajar el modelo. Un nivel superior o global en donde independientemente de las circunstancias, la interacción del modelo E-R encuentra una estrategia de juego para intentar ganar el juego, y un nivel inferior al primero o personal en donde gracias a la percepción de estímulos particulares en los agentes que conforman el equipo guiado por la IA el modelo entra en acción tratando de resolver el problema actual (si lo hay) pero conservando la estrategia generada en el nivel global o adaptándola si es necesario. Lo que resulta en la adaptación del modelo a un ambiente dinámico, resolviendo situaciones instantáneas y conservando sus objetivos finales.

De esto se deriva una siguiente contribución: Originalmente, el modelo E-R, en su implementación orientada al ambiente literario, trabaja sin la fijación de un objetivo definido, lo que le da paso a su creatividad. En esta investigación, la adaptación del modelo permite creativamente, fijar sus objetivos finales, conservarlos (perseguirlos) y adaptarlos si es necesario.

Una contribución mas, es que en esta investigación, aunque no se tomó en cuenta el parámetro “relaciones” que usa el modelo original para guiar su intervención, aquí se toma en cuenta el factor “actitud” que tiene mas que ver con aspectos puramente humanos. Específicamente, como aportación respecto a este rublo, se dice que, una actitud pudo simularse mediante el manejo del concepto “conveniente” desde el punto de vista de un jugador humano que toma su lugar en una partida en el juego. Este concepto fue manejado muy bien por la implementación del diseño original del modelo, ya que los archivos de experiencia previa guardaban datos que relacionaban la definición de una actitud a una situación en particular. Por lo que el trabajo de búsqueda, comparación, filtrado y transformación de la información inherente a E-R pudo manejar bien este concepto.

Desde el punto de vista del autor, una importante contribución de esta investigación es la implementación de la conexión entre una parte receptora, una parte reactiva y la parte inteligente de este proyecto. Estas tres partes, aunque totalmente diferentes por el propósito que tienen, deben de estar comunicadas entre si, con lo que se vio que debe de haber una estandarización en el manejo de la información, la misma que fue manejada para hacer esta adaptación.

Para lograr la coordinación de estas tres partes debió de haber un módulo encargado de la evaluación constante de la situación del ambiente, de modo que recibido un estímulo o mejor dicho, lanzado un evento en el ambiente, la parte receptora de señales lo capta, obtiene la información necesaria y la envía a la parte inteligente (implementación del modelo E-R) el cual toma una decisión. Esta decisión es administrada y asignada al agente apropiada por el módulo coordinador, y finalmente, la decisión se hace manifiesta cuando la parte reactiva de cada agente entra en acción haciéndose manifiesta en la interfaz gráfica mediante un movimiento o una acción en particular.

6.3 Propuestas para trabajos futuros

En un ambiente dinámico, sería de sumo interés que pudiera mejorarse además de las tensiones que estipula el modelo E-R, el parámetro relaciones. Con certeza este parámetro puede influenciar de manera trascendente la asignación de tareas a los agentes existentes en el ambiente y hacer mucho mas flexible y sobre todo interesante la forma de trabajar del módulo de coordinación entre las tres partes constitutivas que se han diferenciado en este trabajo (percepción, reacción y procesamiento de la información).

Puede además pensarse en que la parte administradora o coordinadora de la implementación del modelo E-R a un ambiente dinámico pudiera ser otra técnica de IA, o lo que es mas, pensar en un tercer nivel de acción del modelo E-R, en donde la entrada sea el cúmulo de información obtenida de la toma de decisiones del mismo modelo y donde la salida sea la asignación de una de esas a los agentes del ambiente.

Todo lo anterior pudiera darle una flexibilidad mayor a la adaptación del modelo Ensimismamiento-Reflexión a un ambiente dinámico.

6.4 Conclusión general

Este trabajo termina estableciendo la conclusión más importante, cumpliendo de esta manera con su objetivo:

La adaptación del modelo Ensimismamiento-Reflexión del proceso creativo de Pérez y Pérez se puede adecuar a ambientes dinámicos con un videojuego simple de estrategia.

Bibliografía

- [1] Acosta Villaseñor E. Raúl. “Aplicación de un Modelo en Computadora del Proceso Creativo a la Solución de Problemas de Geometría”, Tesis de Maestría, IIMAS, UNAM, 2004.

- [2] Bentley P. J.: Is Evolution Creative?, WC1E 6BT, University Collage London, Department of Computer Science, Londres, 2002.

- [3] Boden, M. A.: Creativity. En Artificial Intelligence [6], páginas 267-291.

- [4] Laureano Cruces Ana Lilia y E. P. Gilberto: Behavioral design to model a Reactive Decisión o fan Expert in Geothermal wells, Internacional Journal of Aproximate Reasoning, 2003.

- [5] Pattie Maes: A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature, Massachussets Institute of Technology, AI-Laboratory, Estados Unidos, 2003.

- [6] Pattie Maes: Moldeling Adaptive Autonomuos Agents, MIT, IA-Laboratory, 2004.

- [7] Penagos, Julio y Aluni, Rafael: [Preguntas más frecuentes sobre creatividad](#). Revista Psicología, edición especial sobre creatividad, 2000.

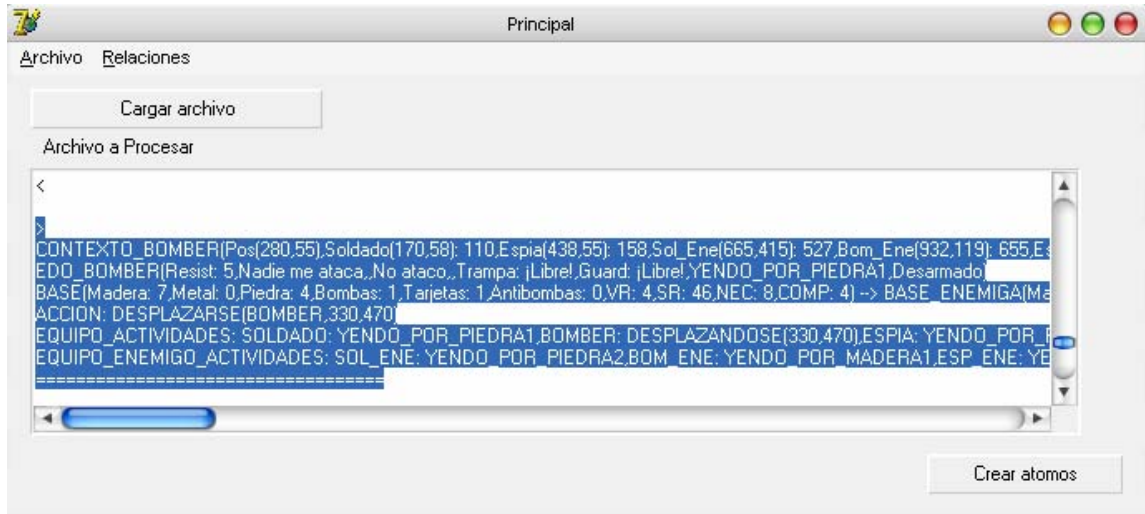
- [8] Pérez y Pérez, Rafael: MEXICA: a computer Model of creativity in Writing. Tesis de doctorado, University of Sussex, Inglaterra, 1999.

- [9] Pérez y Pérez, Rafael y Mike Sharples: MEXICA: A computer model of a cognitive account of creative writing. J. Exp. Theor. Artif. Intell., 13:119-139, 2001.

- [10] Russell Stuart and Norving P., Inteligencia Artificial, Un Enfoque Moderno, 2ª. Ed. 2005.

- [11] Subraya Dasgupta: Is Creativity a Darwinian Process?, Creativity Research Journal, Vol. 16, No. 4, 403-413, 2004.

APÉNDICES



Apéndice B

El programa GeneraMensajes

La implantación del juego Guns esta hecha siguiendo la teoría propia de la IA denominada Agentes Inteligentes. Esta señala que existen entradas específicas para cada agente, quienes las procesan de manera particular dependiendo la naturaleza de su implementación, misma que influye en sus salidas y en su manera de actuar con el ambiente. Por ello se decidió implantar los agentes Inteligentes usando como entradas mensajes definidos por el usuario en un ambiente de programación para el sistema operativo Windows para que puedan ser interpretados correctamente, al igual que sus salidas.

Por estas razones, y por el número total de agentes implementados, el número de mensajes ascendieron a mas de 100, y puesto que se tiene que definir el origen de cada mensaje, el/los emisor(es) y receptor(es) de estos, se tuvo que desarrollar un programa para administrar estos mensajes, lo que resulto en el programa GeneraMensajes. La internaza de este programa es la siguiente:

Msg	Envía	Descripción	Recibe	¿Resp?	Msg	Descripción
1	SOLDADO	Dame Coordenadas Madera uno	MATERIAL	Si	7	Coordenadas de Madera uno
2	SOLDADO	Dame Coordenadas Madera dos	MATERIAL	Si	8	Coordenadas de Madera dos
3	SOLDADO	Dame Coordenadas Metal uno	MATERIAL	Si	9	Coordenadas de Metal uno

Apéndice C

El programa ProcesaExperiencia

Este programa es descrito también en el interior de los capítulos de este trabajo. El propósito es poder automatizar el proceso de estandarización de la información contenida en los comentarios agregados por los usuarios durante la fase de prueba de esta investigación, con la intención de poder procesar esa información por la implementación de la adaptación del modelo Ensimismamiento-Reflexión.

En este programa se cargan los archivos obtenidos en la primera fase de prueba de este proyecto, con la intención de obtener la información que representa la Experiencia Previa en un formato correcto y apropiado para ser manejado en el resto de los procesos implicados en el funcionamiento de la implementación realizada.

Con detalle es descrito este programa durante este trabajo.

Su interfaz es la siguiente:

Apéndice D

El programa AnalizaContextos

Con la intención de poder analizar la información arrojada por el videojuego desarrollado en este proyecto se creo este programa.

En este se tiene una interfaz grafica en donde cargado un archivo que contenga una partida celebrada durante la primer fase de prueba de este proyecto, reproduce la situación al momento de hacer un registro de un agente en el ambiente de juego en tiempo de ejecución.

Gracias al análisis hecho con la ayuda de este programa de pudo determinar gran parte de la información que es relevante en la toma de una decisión por un jugador humano en tiempo de juego.

La interfaz del programa AnalizaContextos se ve en la siguiente figura;

