



**Benemérita
Universidad Autónoma de Puebla**

Facultad de Ciencias de la Computación

**“Comportamientos grupales complejos para
caracteres virtuales y robots”**

Tesis Profesional

Que para obtener el título de:
Ingeniero en Ciencias de la Computación

**Presenta:
José Adán Herrera Walls**

**Asesor:
Dr. Abraham Sánchez López**

Puebla, Pue.

Julio 2009

Resumen

Comportamiento es un concepto al que podemos darle diferentes significados, por ejemplo puede definirse como la acción compleja de un humano, o de otro animal basada por voluntad propia o instinto. En realidad virtual y aplicaciones multimedia puede ser usado como un sinónimo para la animación. Animación literalmente significa “dar vida” ésta puede ser una tarea que consuma mucho tiempo, sin embargo el uso de la computadora puede ayudar a facilitar este proceso. El concepto “animación comportamental” especifica un conjunto de reglas para un personaje permitiendo animarlo de acuerdo a ellas. Reynolds desarrolló la primera animación comportamental con su modelo de comportamientos de flotas donde cada miembro es denominado como “boid” [19].

En este trabajo nuestro objetivo es encontrar una secuencia de movimientos complejos para un conjunto de personajes, se cuenta con una posición inicial y final dentro del ambiente virtual, así como también se conoce el camino que une a estas dos posiciones y que además se encuentra libre de colisión. El resultado es una animación de un comportamiento complejo para un grupo de personajes.

Para llegar a este resultado , investigamos algunos comportamientos simples ya creados, que nos permitan a través de ellos poder obtener comportamientos complejos. Es decir, crear comportamientos complejos a través de comportamientos simples. Los comportamientos complejos pueden incluir más información tanto del ambiente como de los demás personajes, un ejemplo está en los comportamientos que presentamos: comportamiento de huida, persecución, vagar, pasajes estrechos, o seguimiento en círculo.

Índice general

1. Introducción	5
Introducción	5
2. Un análisis de la animación comportamental	7
2.1. Introducción	7
2.2. Mundo virtual	9
2.3. Técnicas de percepción	10
2.3.1. Enfoque por zonas	11
2.3.2. Enfoque sensorial	12
2.3.3. Enfoque por visión sintética	14
2.4. Técnicas comportamentales	15
2.4.1. Enfoque basado en reglas	15
2.4.2. Enfoque de Red	17
2.4.3. Enfoque de inteligencia artificial	17
2.4.4. Enfoque matemático	18
2.5. Un modelo genérico	19
2.5.1. Módulo de variables de estado	20
2.5.2. Regla base	20
2.5.3. Módulo memoria	22
2.5.4. Módulo de movimiento	22
3. Comportamientos para personajes autónomos	23
3.1. Movimiento para personajes autónomos	25
3.1.1. Un vehículo simple	26
3.2. Comportamientos de conducción simples	30
3.2.1. Comportamiento de Búsqueda (búsqueda de un blanco estático)	30
3.2.2. Comportamiento de Huida Simple	30
3.2.3. Comportamiento de Llegada	30
3.2.4. Comportamiento de Evasión de obstáculos	30
3.3. Combinación de comportamientos	31

3.3.1. Comportamiento de Persecución	32
3.3.2. Comportamiento de Evasión	33
3.3.3. Comportamiento Vagar	33
3.3.4. Comportamiento de Huida	34
4. Comportamientos grupales	35
4.1. Comportamiento autoguiado	36
4.2. Comportamiento búsqueda de un objetivo	37
4.3. Comportamiento siguiendo al líder	38
4.4. Pasajes estrechos	40
4.4.1. Formaciones	40
4.4.2. Comportamiento pasajes estrechos	41
4.5. Comportamiento de seguimiento en círculo	44
4.6. Otros comportamientos	46
5. Herramienta RMP3D	47
5.1. Arquitectura de RMP3D	47
5.1.1. Diseño de la herramienta Crowd-Buap	50
5.2. Módulo de Reynolds	56
5.3. Módulo de animación de personajes	56
6. Resultados Experimentales	58
6.1. Entorno de trabajo	58
6.2. Resultados	62
6.2.1. Comportamientos Reynolds	62
6.2.2. Comportamiento Pasajes Estrechos	65
7. Conclusiones y trabajos futuros	71

Índice de figuras

2.1. Un framework para la animación comportamental (estimulo-respuesta).	12
2.2. Framework genérico para la animación comportamental. . . .	21
3.1. Una jerarquía de comportamiento del movimiento.	25
3.2. Fuerzas asimétricas de conducción.	29
3.3. Comportamiento de vagar.	33
4.1. Comportamiento de autoguiado	36
4.2. Comportamiento siguiendo al líder.	39
4.3. Formaciones. (A) Formación en flecha. (B) Formación línea. (C) Formación columna.	41
4.4. Comportamiento de pasajes estrechos. (A) Formación antes del pasaje estrecho. (B) Formación dentro del pasaje estrecho.	43
4.5. Comportamiento de seguimiento en círculo.	45
5.1. Arquitectura de RMP3D.	48
5.2. Caso de uso de la herramienta Crowd-Buap	50
5.3. Subcaso de uso de la herramienta Crowd-Buap	51
5.4. Subcaso de uso de la herramienta Crowd-Buap	52
5.5. Subcaso de uso de la herramienta Crowd-Buap	52
5.6. Diagrama de clases de la herramienta Crowd-Buap	53
5.7. Continuación del diagrama de clases de la herramienta Crowd- Buap	54
5.8. Continuación del diagrama de clases de la herramienta Crowd- Buap	55
6.1. Selección de personaje y de la escena.	59
6.2. El personaje autónomo y su ambiente.	59
6.3. Posición inicial y final del personaje.	60
6.4. Parámetros del Roadmap.	60
6.5. Comportamientos grupales y comportamiento básico disponibles en la herramienta.	61

6.6. Comportamientos Reynolds.	61
6.7. Ejemplo del comportamiento vagar.	62
6.8. Ejemplo del comportamiento de persecución.	63
6.9. Ejemplo del comportamiento de huída.	64
6.10. Ambientes virtuales.	65
6.11. Formaciones columna, flecha, línea.	66
6.12. Cambio de la formación antes de entrar al pasaje estrecho. . .	67
6.13. Personajes dentro de un pasaje estrecho.	68
6.14. Cambio de formación después de cruzar el pasaje estrecho. . .	68
6.15. Ambiente virtual.	69
6.16. Formación en columna.	69
6.17. Todos los personajes vagan dentro de su respectivo ambiente virtual.	70

Capítulo 1

Introducción

Animación literalmente significa “dar vida” ésta puede ser una tarea que consume mucho tiempo, sin embargo el uso de la computadora puede ayudar a facilitar este proceso. En años recientes el concepto “animación comportamental” ha tomado fuerza, este concepto especifica un conjunto de reglas para un personaje permitiendo animarlo de acuerdo a ellas.

El término comportamiento puede tener diversos significados, uno de ellos puede ser la acción compleja basada por voluntad propia o instinto de un humano o de otro animal.

En este trabajo utilizaremos el término comportamiento para referirnos a acciones improvisadas y similares a la vida real de un personaje autónomo. Los personajes autónomos son un tipo de agentes independientes dirigidos a la animación por computadora y medios de comunicación interactivos tales como juegos y realidad virtual. Estos agentes representan a un personaje en una historia o juego y tienen algunas habilidades para improvisar sus acciones.

Reynolds desarrolló la primera animación comportamental con su modelo de comportamientos de flotas (“boids”) [19], le pareció que el proceso de flotas era una actividad descentralizada con cada boid siguiendo su propio conjunto de reglas, ésto es la simulación del movimiento coordinado de un grupo de criaturas. La representación artificial de tales criaturas requiere técnicas para generar el movimiento de entidades individuales dentro del grupo y técnicas para dirigir el movimiento global del grupo.

Este documento de tesis se encuentra estructurado de la siguiente manera:

El primer Capítulo presenta un breve análisis de la animación comportamental, donde además se describen diferentes enfoques que incluyen técnicas de percepción comportamental.

El Capítulo 2 describe lo que es un personaje autónomo y algunos comportamientos para este tipo de personajes como por ejemplo los comportamientos de conducción simples que son el comportamiento de búsqueda, huida, llegada, entre otros, para finalizar se describe el resultado de la combinación de los comportamientos de conducción antes descritos.

Dentro del Capítulo 3 se detallan algunos comportamientos grupales, estos comportamientos están contruidos a partir de comportamientos simples que se basan principalmente en el trabajo de Reynolds.

En el Capítulo 4 se describe la herramienta desarrollada que se implementó. La base una herramienta dedicada a la planificación de movimientos llamada RMP3D, la nueva herramienta se denomina Crowd BUAP, que tiene como finalidad simular comportamientos grupales.

La descripción de algunas pantallas de la herramienta desarrollada se encuentran dentro del Capítulo 5, así como también se muestran algunos de los resultados obtenidos.

Por último en el Capítulo 6 se detallan las conclusiones del trabajo de tesis y se mencionan algunos de los trabajos futuros que pueden seguir la misma línea de investigación que este trabajo.

Capítulo 2

Un análisis de la animación comportamental

Este capítulo presenta un análisis del campo de la animación comportamental. Después de una breve introducción a esta área de investigación, se procede a detallar el uso de varios enfoques. Estos enfoques incluyen las técnicas de percepción, comportamental y de movimiento utilizadas para modelar los personajes animados que interactúan autónomamente dentro de un mundo virtual. Cada enfoque se detalla dentro de una clasificación en la cual las técnicas adoptadas por varios autores se comparan y se contrastan. Concluimos con la propuesta de un framework para la animación comportamental, el cual es una estructura general en el que, todos los sistemas de animación comportamental se pueden modelar y se sugiere que hay una necesidad de un sistema de animación comportamental verdaderamente genérico en el que se puedan modelar todas las clases de comportamientos.

2.1. Introducción

Animación literalmente significa “dar vida”, y ha sido tradicionalmente asociada con animación basada en celdas 2D, es decir, dibujos animados. Las computadoras se han usado para ayudar en la producción de animaciones basadas en celdas, por ejemplo el Rey Leon y Toy Story de Disney. La animación puede ser una tarea que consume mucho tiempo, sin embargo el uso de la computadora puede ayudar a aumentar la velocidad en el proceso de producción. Una de la primeras tecnicas empleadas fue la llamada Keyframing [4], donde el primero y el último frame de la secuencia de una animación son especificados y la computadora genera los frames interiores. En años más recientes, la animación comportamental se ha empleado extensivamente. Es decir, se especifica un conjunto de reglas para un personaje, permitiendo animarlo con este conjunto de reglas. Esto puede ser muy útil en situaciones

en las cuales se desea animar múltiples objetos y estos se comportan de manera similar, por ejemplo una parvada de pájaros. La esencia de este tipo de animación involucra localizar un personaje en un ambiente y permitirle animarse a sí mismo a lo largo del mundo virtual, utilizando su propia respuesta a la estimulación del ambiente y a un conjunto de motivaciones internas.

Inicialmente se desarrollaron modelos basados en física para producir movimientos más realistas. Sin embargo, con estos modelos, cada vez que la animación se ejecuta, los personajes animados se comportarían exactamente de la misma manera. Esto no es un reflejo de la manera en que las criaturas reales actúan y así los investigadores introdujeron modelos comportamentales para tomar en cuenta la individualidad de cada personaje. Reynolds desarrolló la primera animación comportamental con su modelo de comportamientos de flotas (“boids”) [19]. A Reynolds le pareció que el proceso de flotas era una actividad descentralizada con cada boid siguiendo su propio conjunto de reglas. En el sistema de Reynolds [19] para animar parvadas, a cada pájaro se le dio un conjunto de reglas que lo habilitarían para entrar en la animación, imitando la manera en la cual los pájaros reales se reúnen en flota. Describir el movimiento de cada uno de estos pájaros probaría ser una tarea tediosa y que consume tiempo para cualquier animador. Sin embargo, dándole a cada pájaro un conjunto de reglas de flotas a obedecer, los pájaros pueden animarse a sí mismos simplemente sin la necesidad de intervenciones más allá del animador. La motivación del trabajo de los investigadores en animación comportamental proviene del hecho de que le permita al animador estructurar un ambiente, poblado con un número de personajes cada uno con su propio conjunto de reglas comportamentales. Esto permite a la animación tener lugar sin más intervención del animador, lo que efectivamente significa que cada uno de estos personajes animados comportamentalmente es un agente autónomo con su propia percepción, comportamiento, y sistema motor [3]. Estos elementos proveen a la animación comportamental, un nivel de impredecibilidad que significa que en una implementación ideal será casi imposible ejecutar la misma escena un par de veces [22].

El animador puede observar las diferentes interacciones que tienen lugar durante el transcurso de la animación y esto significa que la animación comportamental puede usarse como un vehículo para estructurar simulaciones de eventos del mundo real. Por ejemplo, podría utilizarse para simular el comportamiento de vehículos en un sistema de tráfico. Esto reduce la cantidad de dirección que necesita ser introducida por el animador y la animación resultante puede ser considerada como la simulación visual del modelo comportamental. Se discute a menudo que la animación comportamental comparte sus problemas con aquellos encontrados en inteligencia artificial, robótica, y vida artificial. Por ejemplo, la habilidad de planificar

un camino a través de un ambiente poblado con obstáculos es un problema común en todos estos campos de estudio, incluyendo animación comportamental. Además, el desarrollo de criaturas para uso en ambientes virtuales interactivos será influenciado en el futuro por los avances en animación comportamental [3]. De acuerdo con Reynolds [19], puede ser más correcto llamar a la persona que dirige la animación comportamental como meta-animador, ya que el animador es menos que un diseñador de movimientos y más que un diseñador de comportamiento.

2.2. Mundo virtual

La animación puede tomar lugar dentro de un ambiente particular y así el modelo de un mundo virtual en el cual los personajes animados interactuarán comportamentalmente se crea primero que nada. Cada elemento dentro de la escena necesita ser modelado, incluyendo los personajes animados los cuales toman parte en la animación comportamental. El rendering de los objetos del ambiente involucra crear dos tipos de objetos: objetos estáticos y objetos dinámicos. Los objetos estáticos son aquellos que no requerirán ningún movimiento y así permanecerán estáticos a lo largo de la animación, por ejemplo, una pared. En contraste, los objetos dinámicos se moverán como los personajes animados que forman la animación. En un diseño y creación tanto de objetos estáticos como objetos dinámicos, es la habilidad artística del animador lo más importante y es particularmente útil en la producción de personajes animados.

La creación de un ambiente virtual completo es una combinación de una etapa de modelado por computadora, donde todos los objetos son creados, y el renderizado de esta escena en el cual la animación comportamental tomará lugar. El espectador tiene altos estándares para animaciones que desempeñan los humanos (humanos virtuales). Esto es debido a nuestra habilidad de percibir detalles sutiles del movimiento humano: podemos seleccionar rápidamente acciones improbables o no naturales sin saber exactamente incluso lo que está equivocado. Por lo tanto, la animación de humanos consume particularmente tiempo debido a que necesita capturar las sutilezas del movimiento para transmitir personalidad y humor. Keyframing ha sido el enfoque tradicional para controlar detalles sutiles de humanos virtuales. Sin embargo, la técnica requiere que el animador conozca detalladamente cómo los objetos móviles deben comportarse en el tiempo, además de las habilidades necesarias para producir los Key-frames. Los personajes de la película de 1995 “Toy Story” fueron producidos de esta manera. Una técnica que desarrolla popularidad es la de captura de movimiento. Usa sensores para registrar acciones de un humano en tres dimensiones. Los datos registrados se usan entonces para calcular la animación de un personaje. Esta

aproximación se ha usado en la producción de video juegos de deporte donde la acción de un famoso atleta ha sido capturada por la animación. Los problemas de la técnica incluyen: el potencial de desigualdad entre la forma o las dimensiones del movimiento del sujeto y el personaje animado, restricciones del movimiento del sujeto debido a la presencia de sensores y/o cables, y el cambio de fabricantes y sensores localizados en la piel y ropa durante la ejecución. Todo esto lleva a errores en los datos que introducen acciones no naturales en la parte del humano virtual. Los enfoques más recientes para animar movimientos humanos han sido basadas en simulación. La simulación utiliza las leyes de la física para generar movimientos. La forma, dimensiones e interconexiones de partes del cuerpo se basan en datos biomecánicos y dimensiones de cadáveres. Como los investigadores construyen cada vez modelos más complejos, el resultado de las simulaciones ha llegado a ser cada vez más parecido a la vida real. Los sistemas de control para estos modelos son esencialmente máquinas de estados que seleccionan varios roles que dependen de la conexión en la fase actual de un movimiento. Cada fase tiene asociadas leyes de control para manejar el ángulo de conexión. Simulaciones comparadas con Keyframing y captura de movimiento, se pueden utilizar fácilmente para modificar sutilmente una secuencia de movimientos mientras se mantiene el realismo, por ejemplo acelerar la marcha de un corredor. Además, permiten la interacción con la animación, los personajes pueden responder a las acciones del usuario. La desventaja de la simulación es la especialización y el tiempo requerido para diseñar los sistemas de control, aunque la aproximación de los componentes de software ayuda en esta situación proporcionando una librería de controles prescritos. Sin embargo, producir un humano virtual con toma de decisiones parecidas a las humanas y reacciones, un personaje autónomo, es más complejo que controlar sus movimientos conectados capturados o sus datos de simulación. El personaje debe desplegar una personalidad, habilidad e inteligencia tal como si negociara con el ambiente, situaciones y otros personajes. Para lograr este nivel de desempeño se requieren herramientas para la toma de decisión. Estas herramientas caen en uno de dos niveles: herramientas de bajo nivel que reaccionan tan rápido como es posible al ambiente, por ejemplo en elegir pasos; y herramientas de alto nivel que planifican tareas complejas o ejecutan scripts, por ejemplo planificar una estrategia de búsqueda para localizar un objeto.

2.3. Técnicas de percepción

Las técnicas de percepción son los medios mediante los cuales el personaje animado percibe el ambiente. El tipo de información que puede ser detectado dependerá de la capacidad sensorial del personaje animado. Reynolds

sugiere que es incorrecto proporcionar al personaje información completa y correcta sobre su mundo virtual como conocimiento incompleto, el cual es más parecido al modelo de la realidad. Las técnicas individuales de percepción estarán limitadas dentro de su rango de operación, lo que significa que el personaje animado solo será capaz de percibir su ambiente dentro del alcance de la técnica de percepción que se esté usando. La habilidad de percibir el ambiente dentro de un tiempo razonable es importante, mientras más rápido el personaje pueda detectar objetos en su ambiente, más pronto puede omitir la información sobre su componente de comportamiento, resultando una animación más suave. La elección de la técnica de percepción es importante, ya que el comportamiento del personaje animado dependerá de la calidad de la información que éste reciba. Las técnicas de percepción pueden ser agrupadas en tres:

1. Enfoque por zonas
2. Enfoque sensorial
3. Enfoque por visión sintética.

2.3.1. Enfoque por zonas

El enfoque por zonas considera que el personaje animado es capaz de conocer una o más regiones de percepción alrededor de él y cualquier objeto introducido en esta área será sentido por el personaje. Esto es ilustrado en la Figura 1.1. Reynolds usó la aproximación de zona en su trabajo en un modelo comportamental distribuido, donde cada pájaro, o “boid”, tiene una zona esférica de sensibilidad centrada sobre su origen local y así cualquier objeto introducido en esta zona será detectado por el boid. Usando este tipo de aproximación de zona, Reynolds ha tratado de hacer disponible para el personaje animado aproximadamente la misma información que está disponible para un animal real. Sin embargo, Reynolds no intentó modelar los sentidos usados por un animal real durante la reunión. En lugar de esto ha tratado de dar al personaje animado la información que está al final del resultado del proceso perceptual y cognitivo de un animal real. Reynolds sugiere que una mejora del comportamiento de los boids en los límites de la flota puede ocurrir mediante el uso de una zona ponderada. También señaló que si los boids pudieran ver, serían mucho mejores planificando caminos y el modelo de boid puede ser confundido por simples objetos de tipo laberinto que podrían fácilmente ser distribuidos por un boid con sistema de visión.

El tamaño de la zona de detección tanto en los enfoques de Reynolds y otros autores es importante por dos razones:

1. Una zona muy pequeña puede no ser tan buena para evitar colisiones o para la planificación de caminos como la habilidad de una evasión

de colisiones efectiva, y una planificación de colisiones requerirá posteriormente una cierta cantidad de tiempo, que la zona muy pequeña no puede proporcionar.

2. Una zona muy grande significa más elementos en el proceso que incrementa subsecuentemente el tiempo de cómputo.

De acuerdo a la experiencia de muchos autores, se sugiere que la zona de aproximación pueda ser extendida dando al personaje animado una serie de áreas de detección representativas de un animal real que tiene numerosas regiones en su espacio personal.



Figura 2.1: Un framework para la animación comportamental (estimulo-respuesta).

2.3.2. Enfoque sensorial

El enfoque basado en sensores se refiere a la colocación de sensores virtuales sobre el personaje animado, cada uno opera dentro de una zona de detección específica. Este enfoque permite a diferentes tipos de sensores detectar diferentes cualidades de los objetos percibidos. La elección de los sensores es importante, así como también la orientación y localización de cada uno de ellos. Es a través del estímulo recibido que el personaje adquiere conocimiento acerca de su ambiente.

Se han propuesto una serie de sensores para detectar características específicas de los objetos en el ambiente. Cada sensor en el cuerpo del personaje animado afecta como debe ser estimulado fuertemente. Hay dos tipos principales de sensores:

1. Sensores de distancia, que pueden detectar que tan lejos del personaje está en relación con otros objetos. Lo opuesto son los sensores de

proximidad que detectan que tan cerca está el personaje animado de otros objetos en el ambiente.

2. Sensores de calidad, que detectan cualidades características que son propiedades de otros objetos.

Reich et al. [17] extienden estos conceptos básicos incluyendo más sensores para detectar otras características perceptibles. Crearon un sistema para la locomoción humana en terrenos utilizando razonamiento y han involucrado diferentes tipos de sensores:

- Sensores de objetos: estos proporcionan principalmente la distancia del personaje animado a un objeto particular en el ambiente.
- Sensores de rango: proporcionan principalmente la distancia del personaje animado a un grupo de objetos dentro de un ambiente virtual.
- Atractores: el principal uso de este tipo de sensores es atraer al personaje animado hacia la meta.
- Repulsores: estos sensores son utilizados principalmente en el evitamiento de colisión cuando objetos detectados en el ambiente necesitan ser evadidos.
- Sensores de terreno. Los sensores de terreno evalúan el terreno local dentro del ambiente.
- Sensores de campo: este sensor es usado para detectar regiones visibles a otros personajes animados que son hostiles al personaje.

Una de las limitaciones de estos enfoques sensoriales, es el nivel de realismo dentro de las características perceptibles, así como de las generadas por los objetos en un ambiente virtual. Por ejemplo, en el mundo real, eventos como un sonido pueden solo durar unos segundos. Este tipo de características de percepción han sido modeladas por Maruichi et al. Su trabajo está basado en sensores, cada uno tiene un área accesible o alcance y puede recibir información existente en esa área del ambiente. Los sensores tienen 360° de zona de percepción y esta técnica comparte similitudes con la aproximación por zonas. Cada objeto en el ambiente tiene un área de transmisión que es la zona en la cual el objeto puede ser detectado y un sensor del personaje sólo será capaz de detectar otro objeto cuando el área del sensor intersecta con el área de transmisión del objeto. Este enfoque difiere de los otros en que toma en cuenta la duración de la percepción. Esta técnica es más realista así como la cualidad perceptible es más precisa.

El tamaño de la zona de detección es importante, ya que una zona muy pequeña puede no ser buena para el evitamiento de colisiones mientras que

una zona muy grande significa más elementos al proceso resultante en un incremento de tiempo de cálculo. La habilidad de planificar un camino depende también del tamaño de la zona de percepción, entre más grande sea la zona, mejor será la habilidad de planificar. Todos los modelos de percepción basados en el enfoque de sensores actualmente son muy similares en que todos tienen uno o más sensores para detectar una cualidad de percepción específica. Aunque el tipo de sensores puede variar, cada uno de ellos operará solo dentro de un rango específico.

2.3.3. Enfoque por visión sintética

La visión sintética es una técnica donde el personaje animado es provisto de una visión de su mundo virtual. Esto trae consigo el modelado de una capacidad de percepción que proporciona al personaje animado información que pertenece a los aspectos visuales del ambiente. Esto fue introducido en la animación comportamental por Renault et al. [18] y ellos declaran que desde que la visión de un personaje animado es sintética para el mismo, todos los problemas de reconocimiento de patrones y detección de distancia pueden ser ignorados. Sin embargo como el enfoque de visión sintética sólo será capaz de detectar estímulos visuales, no será de uso en la detección de otros estímulos como lo son olores o sonidos, etc. También propusieron una solución simple y estable que necesita proveer al personaje animado con información de visión sintética de su ambiente. Una ventaja importante de este enfoque es que podemos aprender de la investigación sobre visión humana.

Reynolds ha afirmado, que el mundo de los personajes es un lugar muy brumoso así que la estimulación visual es atenuada por la distancia; sin atenuación, una criatura rodeada por obstáculos está en un ambiente poco interesante en cualquier dirección que apunte, el ve un obstáculo. Éste es un contraste para Renault et al. [18] y Tu et al. [24] ya que sus personajes animados sólo ven dentro de un campo de visión específico. El sistema visual del personaje sólo puede distinguir entre tres tipos de objetos en este mundo (obstáculos, amigos y depredadores).

El tamaño del cono de visión dictará la habilidad del personaje animado para planificar un camino, tan grande sea el campo de visión será mejor la capacidad de planificar un camino y será mejor la capacidad del personaje para evitar colisiones. Esto es debido a que el personaje animado será capaz de detectar el obstáculo más pronto y así tendrá una mejor oportunidad de evitar colisiones. Por otra parte, como las características de percepción de los enfoques de zona y de sensores pueden ser hechos sensibles al contexto en los cuales el caso del cono de visión crece en longitud en un ambiente peligroso y decrece dentro de un ambiente seguro. Si uno puede tomar ventaja del hardware del rendering, puede ser más simple y rápida la manera de extraer

información útil del ambiente [3] y una ventaja importante del enfoque de visión sintética es que es posible aprender de la investigación sobre visión humana [18]. La visión es un sub-sistema muy importante de un personaje animado, es un enfoque ideal para modelar animación comportamental y ofrece una aproximación universal para comunicar la información necesaria del ambiente al personaje en los problemas de búsqueda de un camino y evitamiento de obstáculos [14].

2.4. Técnicas comportamentales

Las técnicas comportamentales proporcionan al personaje animado la habilidad de comprometerse en actividades autónomas, permitiéndole tomar sus propias decisiones. Estas técnicas dan a la animación comportamental su nombre y son básicamente el modelo de toma de decisiones del sistema. El comportamiento puede ser solamente reactivo, una respuesta reflexiva a un estímulo del ambiente de un personaje puede ser una respuesta manejada inteligentemente por los deseos internos y experiencias del personaje animado [13]. La forma de esta respuesta será usualmente un vector de movimiento, aunque podría ser también, por ejemplo un cambio de color o algún otro atributo del personaje. Existe un número de enfoques diferentes que han sido adoptados para la implementación de técnicas comportamentales:

1. Reglas,
2. Redes,
3. Inteligencia Artificial, y
4. Matemáticas.

2.4.1. Enfoque basado en reglas

El enfoque basado en reglas involucra dar a cada personaje animado un conjunto de reglas definiendo como ellos tienen que reaccionar a su ambiente. Cada una de estas reglas es específica a una circunstancia particular del ambiente y por lo tanto muchas reglas serán requeridas para modelar todas las circunstancias posibles. Reynolds [19] modeló el comportamiento distribuido para la simulación de flotas, manadas y parvadas, éste modeló está basado en un enfoque de actores. Pugh [16] describe un actor como un pequeño procesador que está definido únicamente por su comportamiento y está caracterizado por su respuesta al recibir un mensaje. La comunicación entre actores se logra a través de una simple, metáfora uniforme de paso de mensajes. Una característica distintiva del sistema basado en actores es que éste no cuenta con una base de datos centralizada. Más aún el conocimiento es

incluido en actores individuales y de esta manera distribuido a lo largo del sistema. Cada boid dentro de la animación creado por Reynolds debe obedecer a un conjunto de reglas programadas para mantener el comportamiento de la flota.

- El evitamiento de colisiones: ésta se usa para evadir colisiones con compañeros cercanos a la flota.
- Igualación de velocidad: Esto permite al boid intentar igualar su velocidad con los compañeros de la flota.
- Estar en la flota: Esto permite al boids permanecer cercanos a los compañeros de la flota.

Este concepto basado en actores ha sido también adoptado por Haumann et al. [8] quienes han desarrollado un sistema similar a Reynolds [19] en el cual han producido un conjunto de pruebas para crear una librería de actores físicamente comportados para reproducir realísticamente el movimiento de objetos flexibles.

La descripción de las reglas basadas en los sistemas producidos por Reynolds [19] y Haumann [8] sufren de la limitación de que sólo proporcionan reglas específicas para una circunstancia particular. Este problema ha sido diseccionado tanto a Renault et al. [18], como para Noser et al. [14] quienes han basado sus reglas en un autómata local de desplazamiento (DLAs). Estos son básicamente programas que también son usados para lidiar con circunstancias específicas pero probablemente otros enfoques basados en reglas, proporcionan un elemento de utilidad.

El uso de una interfaz establece componentes del modelo comportamental, éste es importante ya que libera al animador de tener que crear aspectos del modelo que de otra manera necesitaría conocimiento de programación, que un animador tradicional puede no tener. Este nivel de interacción con el usuario ha sido también adoptado por Bell [2], en el cual desarrolló un conjunto de programas llamado “pequeño mundo”. Éste proporciona a un mundo poblado generado por computadora de criaturas aparentemente autónomas que interactúan. Las respuestas de las criaturas a otras son determinadas refiriéndose a las características comportamentales introducidas usando un programa diseñador de especies. Los comportamientos implementados se basan en objetos llamados mecanismos efectuados para filtrar la entrada e identificar los objetos y/o eventos que son relevantes al comportamiento, ya sea por que son importantes para lograr la meta del comportamiento o por que su presencia determina la salida del comportamiento dado del ambiente inmediato de la criatura.

Los diferentes comportamientos son organizados en grupos de comportamientos mutuamente inhibidos llamados grupos de comportamientos y

estos son considerados importantes por que localizan la interacción entre comportamientos que permiten agregar nuevos comportamientos. El tipo de componentes comportamentales dentro de este enfoque da a esta técnica su naturaleza genérica tal como un mejor nivel de sofisticación que cualquier modelo actual.

2.4.2. Enfoque de Red

El enfoque de red involucra crear una serie de nodos interconectados, cada uno describe el tipo de respuesta de comportamiento, estos nodos son creados con procedimientos basados en matemáticas. El enfoque de red no es lo mismo que una técnica de red neuronal, ampliamente utilizada en inteligencia artificial, ya que las redes neuronales son usadas para mecanismos de aprendizaje, por el contrario el enfoque no proporciona estas características. En el trabajo de Wilhelms sobre un enfoque interactivo para animación comportamental, el modelo del comportamiento de los personajes animados en el ambiente se representa como una red de estímulo respuesta. Esta red está compuesta por sensores para detectar estímulos del ambiente, efectores, que son motores para impulsar el objeto y una serie de conexiones entre los dos para mapear estímulos de una respuesta. Estas conexiones forman el modelo comportamental para el personaje. En otras palabras, una vez que el estímulo ha sido sentido causa una señal para ser procesada desde los sensores a través de la red a los efectores que producen la respuesta deseada. Sin embargo, las conexiones consisten de una serie de nodos que pueden alterar la señal que pasa sobre su camino hacia los efectores. Es esta modificación de la señal que genera el comportamiento del personaje.

Este enfoque ha demostrado ser efectivo en situaciones donde el animador es requerido para crear animación comportamental interactivamente. Esto se logra vía una interfaz donde el mapeo entre varios nodos puede ser inicializado y las respuestas comportamentales creadas. Sin embargo, la creación de estos comportamientos puede ser bastante difícil de hacer debido a su contenido matemático. Además el concepto de enfoque de red puede ser bastante difícil de lograr y como tal esta técnica no proporciona medios naturales para expresar las ideas de los animadores.

2.4.3. Enfoque de inteligencia artificial

Este enfoque incluye aplicar técnicas de inteligencia artificial, tal como motores de razonamiento [13] y redes neuronales [20], a la definición de aspectos comportamentales del personaje animado.

Una técnica de red neuronal se ha propuesto como posible solución en este tipo de enfoque, es decir cuando la acción deseada está bien definida. Esto ha sido usado por Ridsale [20] en su sistema para modelado conexio-

nista de habilidad dinámica e involucra entrenamiento de la red neuronal para entender la habilidad particular siendo expuesta a un número razonable de ejemplos. El ejemplo dado por Ridsale [20] es un intento de enseñar a un actor sintético a jugar handball. La técnica de pizarrón es flexible y fácilmente extensible y sus capacidades son limitadas solo por las bases de conocimiento y movimientos procedurales que han sido implementados y no por la arquitectura de sí mismo. En contraste, la técnica de red neuronal es intuitivamente razonable ya que pone a la par la manera en que actualmente las personas aprenden nuevas habilidades. Sin embargo, los animadores pueden no encontrar uso de una red neuronal muy intuitiva. El enfoque de inteligencia artificial provee de la habilidad de generar comportamientos que no han sido explícitamente programados y de esta manera da un mejor rango de respuestas comportamentales. Sin embargo, requiere de un cierto nivel de entendimiento del animador acerca de las técnicas a ser usadas.

2.4.4. Enfoque matemático

Esta es una técnica donde el comportamiento del personaje es definido en términos matemáticos y el tipo de enfoque ha sido tomado por Takeuchi et al.[21] y Lethbridge et al. El trabajo de estos autores, involucra un modelo de atracción en el cual un actor decide aproximarse a un objeto dependiendo de si le agrada o le disgusta. Los aspectos comportamentales de este modelo de atracción están compuestos por tres modelos:

1. Un modelo de la naturaleza del objeto: el modelo de la naturaleza del objeto está compuesto por un vector de elementos, donde cada elemento forma una componente de la naturaleza. Por ejemplo, naturaleza = (N_1, N_2) , donde N_1 y N_2 son constantes numéricas.
2. Un modelo de observador de agrado: el modelo de observador de agrado es también un modelo vectorial que corresponde al modelo naturaleza. Por ejemplo, el vector agrado $L = (2, -3, 7)$ significa que al observador le gusta más el lado del primer elemento de la naturaleza para el valor 2, y el lado menor del segundo elemento de la naturaleza para el valor $-3, 7$. Cada personaje animado en la animación tendrá su propio modelo de agrado.
3. Un modelo de función: es el modelo que actualmente produce la atracción desde la naturaleza percibida del objeto y el gusto del observador. Genera la atracción multiplicando el correspondiente agrado y los elementos de la naturaleza para cada componente del vector, y finalmente suma estos productos para dar el valor de atracción.

El valor de atracción es usado por el personaje animado para decidir si le agrada o disgusta un objeto particular y basado en esto, se moverá más cerca, o más allá del objeto percibido. Este modelo de atracción es limitado, ya que solo proporciona para una planificación de caminos, una restricción del comportamiento del personaje a decidir si le agrada o disgusta el objeto percibido y así moverse hacia o lejos del objeto.

Bécheiraz y Thalmann también propusieron un modelo matemático para describir la comunicación no verbal entre actores virtuales. El modelo que representan permite al actor reaccionar a la actitud e interacción social con otros actores. Por ejemplo, dos personas dentro de una conversación íntima a menudo están directamente una frente a otra. Sin embargo, cuando una persona empieza a encontrar el material sujeto no interesante, probablemente se irá. Aunque el mismo modelo comportamental es usado para todos los actores, este es proporcionado a cada uno con una personalidad individual para impedir un grupo homogéneo de comportamientos. Esta personalidad consiste en un valor numérico indicando la personalidad del personaje, estado emocional y relación con otros actores. El modelo comportamental consiste de un número de funciones que operan sobre este valor numérico para determinar las respuestas del personaje a eventos particulares. Las funciones son usadas también para actualizar las relaciones interpersonales entre actores, de este modo les permite evolucionar cuando son afectados por las interacciones. En conjunto el enfoque matemático proporciona un medio de respuestas de especificación comportamental de una manera precisa y, como tal, da a este método la ventaja de permitir la descripción ambigua de comportamientos. Sin embargo, el énfasis en matemáticas significa que puede no proporcionar una manera natural de pensar para los animadores.

2.5. Un modelo genérico

En las bases antes revisadas, tal parece que los sistemas de animación comportamental pueden ser modelados de acuerdo al framework de animación presentado en la Figura 1.2 ésta comprende cuatro modelos de comunicación. El primero de estos es el modelo ambiente, en otras palabras el mundo virtual, que consiste de todos los objetos en la escena animada, incluyendo los personajes comportamentalmente animados. El ambiente de un personaje en particular contiene todos los objetos, incluyendo todos los otros personajes animados cada uno tendrá una percepción, comportamiento y modelo motor. Estos personajes usan sus sentidos, a través del uso del modelo de percepción, para llegar a ser conscientes de su ambiente. Este modelo de percepción consiste de una o más técnicas de percepción y la información sensorial es emitida al modelo comportamental del personaje, que consiste de un mecanismo, esto es una técnica comportamental, para

determinar como el personaje reaccionara a su ambiente, el modelo motor generará un movimiento para el personaje, basado en una de las técnicas motor y el modelo recibe un requerimiento para la salida del modelo comportamental. Ya que el personaje es también parte del ambiente, cuando da una respuesta motor, el modelo ambiente debe ser actualizado adecuadamente. Este tipo de framework es la base de una arquitectura más formal presentada en la Figura 1.2. El modelo comportamental se comunica con los modelos de percepción y motor vía las plantillas de interfaz. Su propósito es facilitar la comunicación con cualquier modelo percepción o motor. El módulo de intervención de usuario proporciona al animador el acceso al modelo comportamental. Esto facilita la operación tal como los personajes animados son definidos como “hambrientos” o nuevas metas sean establecidas así como los progresos de la animación. El modelo comportamental es descrito con mas detalle en las siguientes secciones.

2.5.1. Módulo de variables de estado

El “módulo de variables de estado” guarda valores de datos relacionados con la percepción actual del ambiente (variables de percepción), incluyendo su estado mental. Un estado mental de un personaje guarda información con respecto al estado psicológico actual del personaje, por ejemplo, su nivel de miedo o grado de pánico, mientras que las variables de actor representan el estado del personaje dentro del ambiente, por ejemplo, su velocidad y dirección actual. Dentro del estado mental, los disparadores mentales son reglas que son asociadas a una variable de estado mental específica. Estas reglas pueden también tener un timer mental asociado, para facilitar con ellos, por ejemplo, un personaje llega a estar mas hambriento con el progreso de la animación.

2.5.2. Regla base

Estas reglas definidas por el usuario especifican como el personaje reaccionará a estímulos externos, las reglas serán presentadas en forma de sentencias “si-entonces”, en las cuales el valor de algunas variables de estado son examinadas y la reacción es inducida dependiendo en su valor. Cada regla tendrá incluido un “nivel de prioridad” y “un timer de búsqueda de la meta”. La regla con la mayor prioridad será ejecutada y la meta establecida será localizada en la búsqueda de la meta del “módulo de memoria”. Permanecerá ahí por una duración especificada por el valor del timer de búsqueda de la meta, que puede ser definido como indefinitivo.

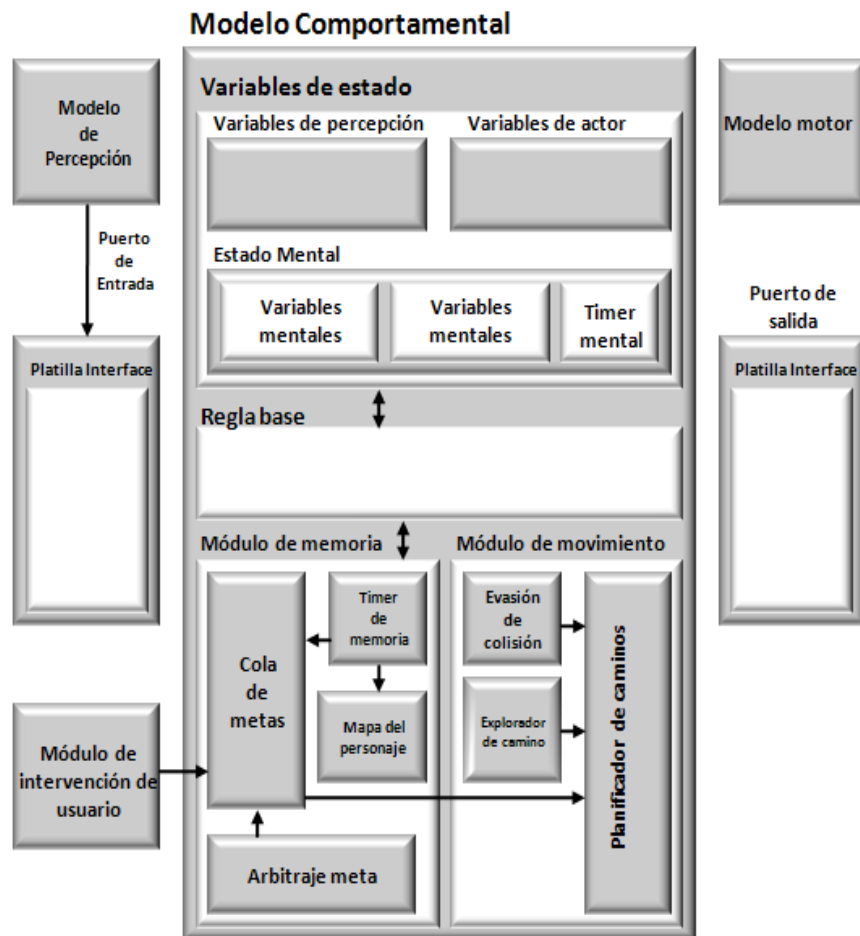


Figura 2.2: Framework genérico para la animación comportamental.

2.5.3. Módulo memoria

El “módulo memoria” es usado para proporcionar al personaje la habilidad de recordar cualquier actividad que aún tiene un desempeño (cola de la meta) o la presencia de otros objetos en su ambiente (mapa del personaje). El módulo de arbitraje de la meta es llamado para resolver conflictos entre las metas y la cola de la meta. Las metas son colocadas en la cola cuando una regla en la base de reglas o disparadores mentales es ejecutado o cuando un animador las coloca directamente. El timer de la memoria puede ser usado para controlar que tanto estas metas permanecerán en la cola.

2.5.4. Módulo de movimiento

El “módulo de movimiento” tiene que ver con todos los aspectos de movimiento del personaje animado, desde generar un plan inicial hasta colisiones. Su operación interna involucra evitación de colisión y resolución así como exploración de un camino y módulos de planificación de caminos. La salida del módulo de movimiento es un conjunto de comandos de movimientos genéricos que son entonces trasladados a los comandos nativos del módulo motor siendo usados vía la interfaz de la plantilla de salida.

Capítulo 3

Comportamientos para personajes autónomos

Los personajes autónomos tienen la habilidad de navegar alrededor de su mundo como en la vida real y con problemas inesperados. Estos comportamientos de manejo o de conducción son en gran parte independientes de los datos usados en las formas de movimiento del personaje, para lograr estos objetivos se utilizan combinaciones de ambientes direccionales (por ejemplo: irse de aquí hacia allá mientras se evitan obstáculos, seguir el pasillo, unir un grupo de personajes).

Los *personajes autónomos*¹ son un tipo de agentes autónomos dirigidos a la animación por computadora y medios de comunicación interactivos tales como juegos y realidad virtual. Estos agentes representan a un personaje en una historia o juego y tienen algunas habilidades para improvisar sus acciones.

El enfoque de los personajes autónomos es contrastante, puede ser un personaje en una película animada, en cuyo caso las acciones están establecidas de antemano, o un avatar¹ en un juego o en un sistema de realidad virtual, en este caso las acciones se toman en tiempo real por un jugador o participante. En juegos, los personajes autónomos son algunas veces llamados *personajes no controlados*.

Un personaje autónomo debe combinar aspectos de un robot autónomo con algunas herramientas de un actor humano en un escenario improvisado. Estos personajes usualmente no son robots reales, y ciertamente tampoco actores humanos, pero comparten características de ambos.

¹Personalidad virtual que puede adoptar el usuario de un chat o juego de rol virtual, que permite cambiar de sexo, de raza o edad.

El término “agente autónomo” es usado en muchos contextos, en relación a otros campos de estudio. Un agente autónomo puede existir en aislamiento, o éste puede ser *situado* en un mundo compartido por otras entidades. Un agente “data mining” es un ejemplo del primer tema, y un controlador de una celda de poder es un ejemplo del último. Un agente *situado* puede ser *reactivo* (instintivamente, manejado por estímulo) o éste puede ser *deliberativo* (“inteligente” en el sentido clásico de la inteligencia artificial). Un agente autónomo puede tratar exclusivamente con información abstracta (“softbot”, “knowbot”, o “agente de información”) o éste puede ser *incrustado* en una manifestación física (un típico robot industrial o un vehículo autónomo). Las combinaciones de *situado*, *reactivo*, e *incrustado* definen distintas clases de agentes autónomos.

En la categoría de *situado*, los agentes *incrustados* usualmente se refieren a robots autónomos: dispositivos mecánicos que existen en el mundo real. Algunas veces los robots son estudiados a través de simulación computacional, pero estas acciones no son bien vistas por los conservadores en el campo de la robótica porque la simulación puede divergir de la realidad en formas no previstas. Hay otras clases de agentes *situados*, los agentes *incrustados* basados sobre un modelo computacional. El término virtual se utiliza para denotar a los agentes, los cuales, lejos de ser simulaciones de un dispositivo mecánico en un mundo real, son de hecho agentes verdaderos en un mundo virtual. Los personajes autónomos son: *agentes situados*, *incrustados*, *reactivos* y *virtuales*.

El término *comportamiento* tiene muchos significados. Puede significar la acción compleja de un humano o de otro animal basada por voluntad propia o instinto. También puede significar en gran parte acciones preestablecidas de un simple sistema mecánico, o la acción compleja de un sistema caótico. En realidad virtual y aplicaciones multimedia, esto algunas veces es usado como un sinónimo para la animación. En este trabajo utilizaremos el término *comportamiento* para referirnos a acciones improvisadas y similares a la vida real de un personaje autónomo.

Para entender el comportamiento de un personaje autónomo, se realiza una división en varias capas. La Figura 2.1 muestra una división del comportamiento del movimiento para personajes autónomos dentro de una jerarquía de tres capas: *selección de acción*, *dirección*, y *movimiento*. Existen otras subdivisiones posibles, una jerarquía similar de tres capas es descrita por Blumberg y Galyen [3], ellos llaman a las capas: *motivación*, *tarea*, y *motor*.

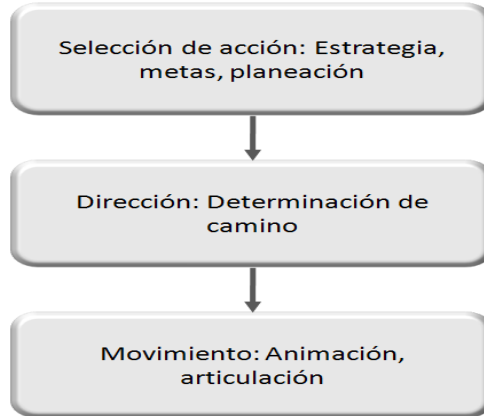


Figura 3.1: Una jerarquía de comportamiento del movimiento.

Consideremos, por ejemplo, algunos vaqueros dirigiendo a una manada de ganado fuera de la montaña. Una vaca se aparta del rebaño. El jefe pide a un vaquero recoger al animal. El vaquero rápidamente gira su caballo y lo guía hacia la vaca, posiblemente esquivando obstáculos a través del camino. En este ejemplo, el jefe representa la *selección de acción*: el estado del mundo ha cambiado (una vaca dejó el rebaño) y estableciendo una meta (traerla de vuelta). El nivel de *dirección* está representado por el vaquero, quien descompone el objetivo dentro de una serie de sub-objetivos (aproximarse a la vaca, evitar obstáculos, traerla de vuelta). Un sub-objetivo corresponde al comportamiento de direccionamiento para el equipo formado por el caballo y vaquero. Usando varias señales de control (comandos vocales, espuelas, riendas) el vaquero dirige su caballo a través del blanco. El caballo implementa el nivel de *movimiento*. Tomando las señales de control del vaquero como entrada, el caballo se mueve en la dirección indicada. Este movimiento es el resultado de una interacción compleja de la percepción visual del caballo, su sentido de equilibrio, y de sus músculos aplicando movimiento a su esqueleto. Desde un punto de vista de ingeniería, el movimiento de las piernas es muy fuerte [29][18], pero ni el vaquero ni el caballo dan a esto alguna importancia.

3.1. Movimiento para personajes autónomos

El movimiento es la parte inferior de los tres niveles de la jerarquía de comportamiento del movimiento descrita arriba. La capa de movimiento representa una encarnación del personaje. Éste convierte sus señales de control de la capa de direccionamiento dentro del movimiento del cuerpo del per-

sonaje.

Recordemos, el ejemplo del caballo y el vaquero como representación de la capa de movimiento. Las decisiones de dirección del vaquero son tomadas como una simple vía, como señales de control para el caballo quien las convierte en movimiento. Ahora imaginemos al vaquero fuera del caballo y colocado sobre una motocicleta. El objetivo de la *selección* y comportamiento direccional del *movimiento* recae sobre lo mismo. Todo lo que ha cambiado es la forma de llevar a cabo las señales de control dentro del movimiento. Originalmente esto envolvía movimientos de piernas y ahora esto envuelve movimiento de ruedas. El rol de montador no ha cambiado.

Esto sugiere que, con una convención apropiada para las señales de comunicación de control, los comportamientos de dirección pueden ser completamente independientes del esquema específico de movimiento.

El movimiento de un personaje autónomo puede estar basado en (o independiente a) la representación animada. Un personaje puede estar representado por una simulación de balanceo dinámicamente basada en la manera física de caminar, proporcionando a ambos animación realista y movimiento comportamental. O un personaje puede tener un modelo muy simple de movimiento donde se agrega una representación estática o pre-animada. Una aproximación híbrida es el uso de un modelo simple y un modelo adaptativo animado y colocar un puente entre la abstracción del movimiento y el terreno concreto. Finalmente, el movimiento puede ser restringido a la herencia de movimiento en un conjunto arreglado de segmentos pre-animados, los cuales son seleccionados discretamente o mezclados.

3.1.1. Un vehículo simple

Utilizaremos un modelo simple de movimiento para introducir la discusión de comportamientos direccionales. Este modelo de movimiento está basado en un vehículo ideal simple, este modelo de vehículo es una aproximación de una masa puntual, por una parte, esto permite simples y pocas operaciones del modelo basado en física (por ejemplo, una masa puntual tiene velocidad (momento lineal) pero ningún momento de inercia (momento rotacional)). Por otra parte, esto no puede ser un modelo físico completo porque los puntos de aproximación no existen en el mundo real. Cualquier objeto físico con masa tiene un radio distinto de cero y así un punto de inercia. Se usa un modelo simplificado no basado en física por conveniencia y se intenta representar uno (físicamente) con cierta “perdida de generalidad”.

Una masa puntual está definida por *una propiedad de posición y una propiedad de masa*. El modelo simple vehicular incluye también *una propiedad de velocidad*. La velocidad está modificada por fuerzas aplicadas. Como es un vehículo, estas fuerzas son generalmente autoaplicadas, y por lo tanto simplificadas. Por ejemplo, una fuerza típica la cual ajusta un vehículo a una velocidad de empuje, generada por la misma fuerza de poder del vehículo, y por lo tanto se limita su magnitud por la capacidad de su fuerza de poder. Para el modelo vehicular simple, esta noción está ejemplificada por un simple parámetro de “fuerza máxima” (*fuerza_máxima*). Típicamente esta limitación se debe a la interacción entre la aceleración debida al empuje y la des-aceleración debida al roce, fricción viscosa o partes recíprocas. Como una alternativa a la simulación realista de estas fuerzas limitadoras, el modelo vehicular simple incluye un parámetro de “velocidad máxima” (*velocidad_máxima*). Este límite de velocidad esta limitado por una truncación cinemática del vector de velocidad del vehículo. Finalmente, el modelo vehicular simple incluye una “orientación”, la cual toma la posición del vehículo en forma de coordenada vehicular en el espacio, donde el modelo geométrico del vehículo puede ser agregado.

Para un modelo de un vehículo en 3D, el vector *posición y velocidad* tienen tres componentes y el valor de *orientación* es un conjunto de tres vectores. Para un vehículo en 2D, cada vector tiene dos componentes, y el valor de la *orientación* tiene dos vectores base o puede ser representado como un escalar simple de ángulo simple.

La física del modelo de un vehiculo simple está basada en una integración de Euler. A cada paso de la simulación, se determina el comportamiento de las fuerzas de manejo (limitadas por la *fuerza_máxima*) que son aplicadas a la masa vehicular puntual. Ésta produce una aceleración igual a una fuerza de manejo dividida por la masa del vehículo. La aceleración es agregada a una velocidad anterior que produce una nueva velocidad, la cual está truncada por la *velocidad_máxima*.

El modelo de un vehiculo simple mantiene su espacio local de velocidad por *ajuste incremental* del paso anterior. El sistema de coordenadas local está definido en términos de cuatro vectores: un vector de posición especificando el origen local, y tres vectores de dirección que sirven como vectores base del espacio. Estos vectores base indican la dirección y longitud de unidades coordenadas en cada una de las tres direcciones mutuas perpendiculares mutuas relativas al vehículo. Los ejes se conocen como *forward*, *up* y *side*, (X, Y y Z).

Para permanecer alineado con la velocidad a cada paso, los vectores base

deben ser rotados dentro de una nueva dirección. (Si la velocidad es cero, la orientación anterior es conservada). De hecho usando rotaciones explícitas, el espacio local es reconstruido usando una combinación de sustitución, aproximación y re_ortogonalización. Se inicia con la nueva velocidad y una aproximación para el *nuevo_up*. Por ejemplo, la anterior dirección *up*, puede ser usada como una aproximación al nuevo *up*.

La idea básica es que el *aproximado_up* es cercanamente perpendicular a la nueva dirección de *forward*, porque los cambios de cuadro en cuadro en la orientación son típicamente pequeños. La nueva dirección de *side* será perpendicular al nuevo *forward*, de la definición de producto cruz. El nuevo *up* es el producto cruz de la perpendicular *forward* y *side* y también es perpendicular a cada uno.

El concepto de “*alineación de velocidad*” no únicamente especifica una orientación. El grado de libertad correspondiente a la rotación alrededor del eje *forward* (también conocido como *rol*) sigue siendo libre. Construyendo el nuevo espacio local relativo a uno previo (por ejemplo, usando la anterior dirección *up* como la aproximación inicial del nuevo) asegura que el sentido (*rol*) de la orientación siga siendo al menos constante. Definir el valor “correcto” de la orientación (*rol*) requiere heurísticas más detalladas basadas en el uso intencionado de un modelo vehicular.

Para un vehículo “volador” (avión, nave espacial, y submarinos) usualmente se define un *rol* en términos de *ingreso*. La idea básica de *ingreso* es para alinear el “piso” del vehículo (eje -*up*) con la gravedad aparente debido a la fuerza centrífuga durante un turno. Inversamente se quiere que la dirección de *up* este alineada a la fuerza centrípeta que produce la operación. En la presencia de gravedad, la dirección de *down* se debe alinear con la suma de la aceleración y la aceleración gravitacional. También hay que agregar la orientación actual para salvar los cambios del *rol*. Así para implementar el *ingreso* en el modelo vehicular simple, la dirección aproximada *up* es una suma que considera: la aceleración de manejo, la aceleración gravitacional, y la anterior *up*.

Para una “superficie deslizante” (rodado, resbalando, o a pié) del vehículo, se desea colocar la posición del vehículo en la superficie y alinear el eje *up* del vehículo en la superficie normal. Además la velocidad debe ser obligada a ser puramente tangencial en la superficie. Estos requerimientos pueden ser fácilmente conocidos si la superficie múltiple está representada como un camino, y así un punto arbitrario del espacio (correspondiendo a la posición anterior del vehículo) puede ser representado como: (1) el punto más cer-

cano sobre la superficie, y (2) la superficie normal a ese punto. La velocidad puede ser tangente restando la porción normal a la superficie. La posición del vehículo se fija al punto de la superficie, y la superficie normal llega a ser su eje *up*.

Pasar la señal de control de los comportamientos direccionales al comportamiento del movimiento consiste exactamente de una cantidad vectorial: una fuerza de dirección deseada. Los modelos más realistas del vehículo tendrían sistemas muy diversos de señales de control. Es posible mapear un vector de dirección generalizado a señales escalares: el componente *side* del vector de dirección puede ser interpretado como una señal de dirección, el componente del vector de dirección *forward* puede ser mapeado como la señal de aceleración si es positiva, o como la señal de freno si es negativa. Estos mapeos pueden ser asimétricos, por ejemplo un automóvil típico puede desacelerar debido al frenado mucho más rápido que cuando acelera debido al motor. Ver Figura 2.2.

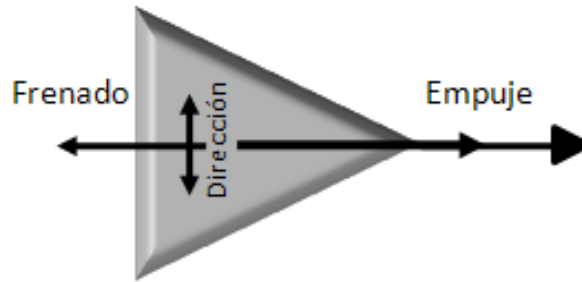


Figura 3.2: Fuerzas asimétricas de conducción.

Debido a la alineación de la velocidad, este modelo vehicular simple no puede simular efectos como derrapes, volteretas, o deslizamientos. Solo permite al vehículo girar cuando su velocidad es cero. Muchos vehículos reales no pueden hacer esto (“no holonomicos”) y en algunos casos esto permite cambios bruscos en la orientación durante un simple instante. Esto se puede resolver colocando una limitación adicional sobre el cambio de la orientación, o limitar el componente lateral de dirección a una baja velocidad, o simulando un momento de inercia.

3.2. Comportamientos de conducción simples

En este capítulo son descritos algunos comportamientos de conducción simples, que servirán como una base para poder realizar comportamientos más complejos, los cálculos utilizados para realizar los movimientos son obtenidos para un solo vehículo, este vehículo es el vehículo simple descrito anteriormente.

3.2.1. Comportamiento de Búsqueda (búsqueda de un blanco estático)

Este comportamiento actúa dirigiendo al personaje hacia una posición especificada en un espacio global. Ajusta al personaje para alinear su velocidad radialmente hacia el objetivo. La “velocidad deseada” es un vector en la dirección del personaje hacia el objetivo. El tamaño de la “velocidad deseada” puede ser *velocidad_máxima* o éste puede ser su velocidad actual, dependiendo de la aplicación específica. El vector dirección es la diferencia entre su velocidad deseada y la velocidad actual del personaje.

3.2.2. Comportamiento de Huida Simple

La huida simple, es en sí, la inversa del comportamiento de búsqueda, ya que actúa sobre la dirección del personaje para alinear su velocidad radialmente lejos del objetivo. Esta velocidad deseada puede dirigirse en una dirección opuesta.

3.2.3. Comportamiento de Llegada

El comportamiento de llegada es la base para construir comportamientos más complejos. En este comportamiento existe solo un agente dentro de un ambiente sin obstáculos, inicialmente el personaje cuenta con una posición inicial y que es generada de manera aleatoria, la posición final es generada a una distancia determinada con respecto a la posición inicial ya creada, por último se generan las posiciones para que el agente alcance la meta.

3.2.4. Comportamiento de Evasión de obstáculos

El comportamiento evasión de obstáculos permite a un personaje la habilidad de manipularse en un ambiente desordenado evitando obstáculos.

La implementación del comportamiento de evasión de obstáculos se pretende que, tanto el personaje y el obstáculo pueden ser vistos como esferas, aunque se podría extender a otras formas diferentes. Imaginemos un avión intentando esquivar una montaña. Ninguno tiene forma esférica, pero

bastará pensar que la esfera unida al aeroplano esquiva a la esfera unida a la montaña.

Una jerarquía de esferas unidas puede ser usada para una representación eficiente de formas para la detección de colisiones [10], e igualmente para la evasión de obstáculos.

Es conveniente considerar la situación geométrica del sistema de coordenadas locales del personaje. El objetivo del comportamiento es mantener un cilindro imaginario en el espacio libre frente del personaje. El cilindro se encuentra sobre el eje delantero del personaje y tiene un diámetro igual a la esfera unida al personaje.

3.3. Combinación de comportamientos

Los comportamientos descritos se pueden ocupar como bloques de patrones para comportamientos más complejos. Para hacer a los comportamientos más interesantes y semejantes a los comportamientos de la vida real necesitamos seleccionar entre ellos, y mezclar, estos componentes individuales. A menos que un personaje autónomo exista en un mundo simple, rara vez tendría sentido que el personaje ejecute continuamente un solo comportamiento de conducción.

Los comportamientos combinados pueden seguir dos caminos. Secuencialmente un personaje puede seleccionar entre modos de comportamiento como cambios circunstanciales en su mundo. Por ejemplo, imaginemos ovejas pastando en un prado cuando repentinamente captan lobos aproximándose. Estos eventos accionan un interruptor del comportamiento discreto. Olvidándose de pastar las ovejas voltean para escapar de los depredadores. No hay tendencia para mezclar estos comportamientos: una oveja no caminará despacio mientras es perseguida por un lobo que corre tras de ésta para devorarla. Estos cambios discretos de estados comportamentales tienen lugar en el nivel de selección-acción, el más alto de los tres niveles en la jerarquía comportamental.

Por otro lado, algunos tipos de comportamientos son comúnmente mezclados, efectivamente en paralelo. Por ejemplo, las ovejas corren a través del bosque, ellas mezclan evasión entre ellas y evasión de obstáculos, para permitirse escapar de los lobos mientras zigzaguean los árboles. Una oveja no puede permitirse ignorar algún componente del comportamiento, ésta siempre se mueve en una dirección para tomar un camino evitando el alcance del lobo y de chocar con los árboles. Esta mezcla de comportamientos ocurre en el nivel medio de dirección de la jerarquía de comportamientos.

La mezcla de comportamientos direccionales puede realizarse de diferentes formas. La más sencilla es simplemente calcular cada uno de los comportamientos direccionales y sumarlos, posiblemente con un factor de peso

para cada uno de ellos. La combinación lineal simple, frecuentemente trabaja bien, pero tiene dos pequeñas desventajas: ésta no es la aproximación calculada más eficiente, y a pesar de ajustar pesos, los comportamientos direccionales pueden eliminar algún otro en tiempos inoportunos.

En ocasiones algunos comportamientos se cancelan con otros, y eso puede resolver tratando de asignar una prioridad a los componentes. (Por ejemplo: primera prioridad es evitamiento de obstáculos, segunda es huida, etc). El controlador direccional verifica primero si el evitamiento de obstáculos no regresa un valor de cero (indicando una potencial colisión), si es así usa éste. De cualquier forma, éste se mueve sobre el comportamiento de segunda prioridad, y así sucesivamente.

Un híbrido para esta técnica es “elegir por prioridad”: con una cierta probabilidad de comportamiento, la primera prioridad es evaluada, y si regresa un valor diferente de cero el valor podría ser usado. De otra manera (si el comportamiento regresa cero) el comportamiento con segunda prioridad es considerado, y así sucesivamente.

En [19] un esquema de mezcla llamado asignación dada la prioridad de aceleración fue usado con el modelo de manadas boids. La idea básica fue que ajustando sus magnitudes, los comportamientos de alta prioridad pueden decidir si dejan o no alguna fuerza direccional para ser usada por comportamientos de baja prioridad.

3.3.1. Comportamiento de Persecución

Este comportamiento es similar al comportamiento de búsqueda excepto que el objetivo es otro personaje en movimiento. Una persecución eficaz requiere una predicción sobre la posición futura del objetivo. Lo que se necesita aquí es usar un predictor simple y una reevaluación sobre cada paso de la simulación. Por ejemplo, un predictor basado en la velocidad lineal corresponde en asumir que el objetivo no girará durante el intervalo de predicción. La posición de un personaje T en una unidad de tiempo en el futuro (asumiendo que no lo maniobra) puede ser obtenida por escalamiento de su velocidad por T y agregando el desplazamiento a su posición actual. El direccionamiento por persecución es simplemente el resultado de aplicar el comportamiento de búsqueda hacia la localización predicha del objetivo. La clave de esta implementación de persecución, es el método usado para estimar la predicción del intervalo T . Idealmente, T sería el tiempo de intercepción, pero el valor es desconocido porque el objetivo puede ser arbitrario y con representaciones no predichas. T puede asumirse constante, mientras que esta asignación ingenua, produciría una mejor persecución que una simple búsqueda (cuando corresponde a $T = 0$). Sin embargo, el funcionamiento razonable de T debe ser más grande cuando el buscador está lejos del objetivo, y pequeño cuando son próximos. Una estimación simple de la calidad

moderada es $T = Dc$ donde D es la distancia entre el buscador y el objetivo, y c es el parámetro de cambio. Un estimador más sofisticado puede ser obtenido considerando los encabezados relativos del buscador y del objetivo, y si el buscador está generalmente delante, detrás, o al lado de el objetivo.

3.3.2. Comportamiento de Evasión

Evasión es un comportamiento análogo al de persecución, en este caso se utiliza el comportamiento de huida para dirigirse lejos de la posición futura predicha del personaje objetivo. Existen técnicas optimas para la persecución y evasión en el campo de la teoría de control [11]. En sistemas naturales, la evasión es a menudo "intencionalmente" no óptima para ser impredecible, permitiendo una predicción de estrategias de persecución [5].

3.3.3. Comportamiento Vagar

El comportamiento Vagar, es un tipo de direccionamiento aleatorio. Una implementación sería generar una fuerza de direccionamiento aleatoria, esto produce un movimiento no necesario. Una mejora de la idea anterior es retener el estado de la dirección y hacer pequeños movimientos aleatorios. Esta idea puede ser implementada por varias formas, pero una que produce buenos resultados es limitar la dirección establecida levemente en la parte frontal del personaje. La producción del direccionamiento para el siguiente cuadro, es un desplazamiento aleatorio. Esto es, generar puntos aleatorios, cada punto generado deberá ser considerado como una submeta por el personaje, de esta manera el personaje visitará varias localidades dentro del ambiente y la dirección del personaje toma una "marcha aleatoria" de una dirección hacia otra.

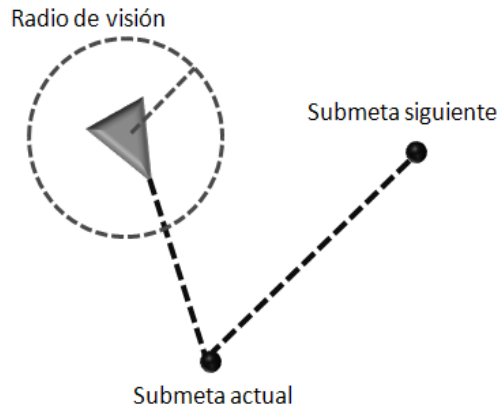


Figura 3.3: Comportamiento de vagar.

3.3.4. Comportamiento de Huida

En este comportamiento los personajes mantienen un papel constante de seguidor y evasor en este comportamiento, así también se mantienen a una cierta distancia. En la implementación de este comportamiento se utilizan el enfoque de esferas utilizado en el comportamiento de evasión de obstáculos, con la diferencia de que las esferas se encuentran en cada personaje, simulando un radio de visión en cada personaje. Mientras el seguidor no se encuentre dentro del radio de visión del evasor, éste se mantendrá en el último punto conocido libre de colisión. En caso contrario, el evasor utilizará el comportamiento de huida simple, con objetivo el seguidor.

Capítulo 4

Comportamientos grupales

La habilidad para simular el movimiento coordinado de un grupo de criaturas juega un papel importante en la vida artificial. La representación artificial de tales criaturas requiere técnicas para generar el movimiento de entidades individuales dentro del grupo y técnicas para dirigir el movimiento global del grupo. Desafortunadamente los métodos existentes no se adecuan correctamente, si se requiere de una navegación compleja. Por otro lado, los algoritmos de planificación de caminos desarrollados en la comunidad de Robótica, son capaces de navegar en ambientes complejos [12].

En particular, el problema de la planificación de movimientos se soluciona típicamente en el espacio de configuración libre de colisión. El paradigma más popular para la planificación de movimientos basado en muestreo es el método de Roadmaps Probabilísticos (PRM). La idea central del método PRM es distribuir aleatoriamente un conjunto de nodos en un espacio de configuración libre de colisión y después conectar estos nodos usando un planificador local simple para formar un grafo conocido como roadmap.

Una característica importante de un roadmap es que proporciona una buena aproximación de la conectividad del espacio de configuraciones libre de colisión. Si el roadmap captura correctamente la conectividad del espacio libre, la planificación de movimiento se reduce a una búsqueda en el grafo o árbol creado.

Aunque existen algoritmos de planificación de caminos muy poderosos, estos sólo han sido usados para planificar o trazar los caminos para un sólo objeto en movimiento, es decir, no han sido usados para el comportamiento de un grupo.

4.1. Comportamiento autoguiado

El comportamiento autoguiado consiste en dos submodelos, uno representa el comportamiento individual de los miembros del grupo y el otro influye en el comportamiento global del grupo. En este comportamiento los miembros individuales deben:

- Evitar la colisión con los otros miembros vecinos del grupo.
- Permanecer cerca de sus vecino. El vecindario está definido por una distancia.

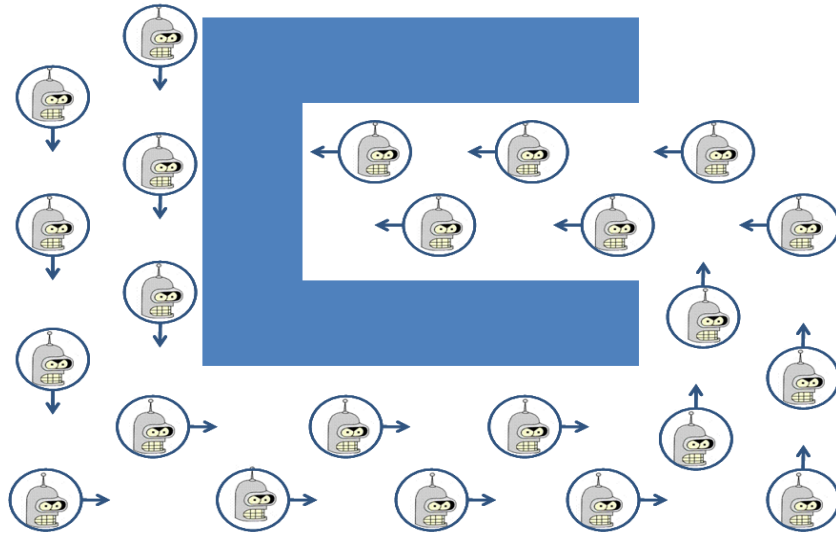


Figura 4.1: Comportamiento de autoguiado

Con la ayuda del planificador se tiene una vista global, se genera el camino entre los dos puntos, una vez que el roadmap es generado, encontrar nuevos caminos es rápido y eficiente, y a partir de este momento un miembro del grupo busca seguir el camino. Para esto, la trayectoria es discretizada en submetas basadas en el rango de sensor de un personaje que es miembro del grupo. Cada miembro guarda el camino de las submetas y tan pronto como un submeta está dentro del rango del sensor de la meta del personaje, la siguiente submeta es la que conduce la dirección de la meta global, es decir, se convierte en la siguiente meta de ese personaje. Así, todos los personajes se trasladan juntos mientras que se mueven hacia la meta. Esto lleva al grupo hacia la meta sin quedar atrapados en un mínimo local [9].

Esto se describe en el siguiente algoritmo:

Algoritmo autoguiado

```
Crear Posición inicial para cada miembro
Dividir camino en submetas
Guardar en cada miembro el camino dividido en submetas
Mientras (No se encuentre en la posición final)
  Para (Cada miembro del grupo)
    Si (La meta está en el rango de visión)
      Quedarse cerca de la meta
    Si_no
      Si (La submeta actual está en el rango de visión del miembro)
        La meta del miembro se fija con la submeta siguiente
      Si_no
        Sigue el camino hacia la meta actual
      Fin_si
    Fin_si
  Fin_para
Fin_mientras
```

4.2. Comportamiento búsqueda de un objetivo

Éste es un tipo de comportamiento de exploración. Tenemos que el ambiente es conocido y el objetivo de este comportamiento es buscar una meta y moverse hacia ella. Para realizar esto utilizamos el grafo del roadmap con pesos adaptativos en los nodos. Cada miembro se comporta independientemente del grupo y usa el roadmap para vagar alrededor de él. Específicamente, los personajes siguen los arcos del roadmap y aquí no existen rutas predefinidas. Si ellos alcanzan un nodo del roadmap con varios arcos, estos eligen probabilísticamente un arco del roadmap dependiendo del peso de dicho arco.

Este comportamiento es similar a la optimización de la colonia de hormigas. Ya que los miembros individuales el grupo conocen el ambiente, pero ellos no conocen la localización de la meta. Si un personaje alcanza la localización de la meta dentro del rango de su sensor, su localización es comunicada a los otros personajes del grupo.

Cuando un personaje atraviesa un camino, él recuerda la ruta que ha tomado. Entonces cuando alcanza la meta, se incrementa el peso de los arcos en la ruta tomada. Si el personaje alcanza un nodo del roadmap sin alguna conexión saliente el peso de los arcos que ha seguido será disminuido.

Del mismo modo se decrementarían el valor de los pesos si existiera algún ciclo en el roadmap, en nuestro caso, el personaje no encontrará un arco con algún ciclo, ya que en el roadmap que utilizamos no existe esto [9].

El comportamiento se resume en el algoritmo siguiente:

Algoritmo búsqueda de un objetivo

```

Crear Posición inicial para cada miembro
Mientras (Algún personaje no se encuentre en la posición final)
  Para (Cada miembro del grupo)
    Si (La meta está en el rango de visión)
      Incrementar los pesos de los arcos de la ruta a la meta
    Si_no
      Si (Encontró algún nodo sin conexión)
        Quitar nodos del camino encontrado hasta que una nueva
        liga se encuentre
        Decrementar los pesos de los nodos que se eliminaron
      Si_no
        Seleccionar el nodo vecino del nodo actual de menor peso
        Colocar este nodo en el camino
      Fin_si
    Fin_si
  Fin_para
Fin_mientras
Comunicar a los demás personajes la ruta encontrada
Para (Cada miembro del grupo a excepcion del que encontró la meta)
  Seguir el camino encontrado
Fin_para

```

4.3. Comportamiento siguiendo al líder

Este es un tipo de comportamiento de seguimiento, consiste en designar a un personaje como líder y designado éste, los personajes restantes deberán seguir al líder, esto se realiza con las ideas de los comportamientos explicados anteriormente, ya que se pueden mezclar para obtener un nuevo comportamiento.

Utilizamos el comportamiento de autoguiado para el movimiento del líder, ya que se discretiza el camino encontrado por el planificador y este camino es el que debe seguir el líder para ir del punto de inicio al final, es decir, la idea básica del comportamiento de autoguiado. Para cada uno del

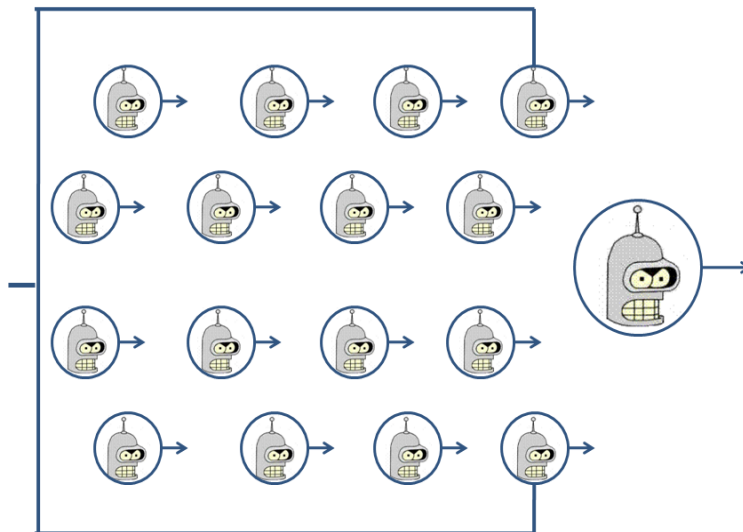


Figura 4.2: Comportamiento siguiendo al líder.

resto de los agentes se utiliza el comportamiento de búsqueda de un objetivo, en lugar de la posición del nodo que debe seguir, se coloca la posición actual del líder, que sigue el camino encontrado, de esta manera, el líder se mueve por el camino encontrado y los personajes lo siguen detrás de él.

Algoritmo siguiendo al líder

```

Crear Posición inicial para cada miembro
Elegir líder
Guardar la trayectoria general en el líder
Mientras (Líder no encuentre la meta)
    Si (La meta está en el rango de visión del líder)
        Quedarse cerca de la meta
    Si_no
        Si (La submeta actual está en el rango de visión del líder)
            La meta del líder se fija como la submeta siguiente
        Si_no
            Caminar hacia la meta actual
        Para (Cada uno de los restantes miembros del grupo)
            Colocar como submeta la posición del líder
            Caminar hacia la submeta actual
        Fin_para
    Fin_si
Fin_mientras
  
```

4.4. Pasajes estrechos

Algunas veces los comportamientos en grupo pueden depender del ambiente que los rodea, el comportamiento de pasajes estrechos surge de esta idea. En el caso de tener un área abierta, existen diferentes formaciones o comportamientos que pueden utilizarse sin mayor problema, sin embargo cuando se trata de pasajes estrechos existen algunas dificultades.

4.4.1. Formaciones

Para poder construir el comportamiento de pasajes estrechos tenemos que entrar a un tema abordado en [23] en donde, se crea un comportamiento de formación para hacer que un equipo de robots sean capaces de alcanzar su meta, evitar peligro y simultaneamente conservar su formación.

Se consideraron diferentes tipos de formaciones para este grupo como son, formación en línea, formación en columna y formación en flecha. A continuación describiremos cada una de ellas:

- **Formación en línea.** En esta formación el grupo viaja en una línea perpendicular a la trayectoria.
- **Formación en columna.** En esta formación se observa como el grupo viaja uno tras de otro.
- **Formación en flecha.** En esta formación el grupo viaja en forma de una “V” invertida.

En la Figura 3.3 se ilustran las formaciones descritas en el parrafo anterior:

La tarea de cada elemento en el grupo en cualquier formación, es llegar a la meta tratando de mantener su posición dentro del grupo. Para que un elemento del grupo pueda mantenerse en la formación se realizará un proceso de dos pasos, que se detalla a continuación:

1. Detectar su punto dentro de la formación: esto es calcular el punto de la formación correspondiente tomando en cuenta los datos proporcionados por el ambiente.
2. Generar una instrucción que indique la dirección correcta del punto donde será ubicado el elemento.

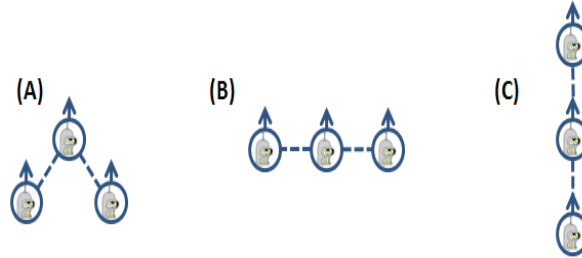


Figura 4.3: Formaciones. (A) Formación en flecha. (B) Formación línea. (C) Formación columna.

Se han identificado tres maneras para determinar el punto de la formación para cada integrante del grupo:

- **Centro unitario de referencia.** Se calcula un centro unitario independientemente de cada elemento dependiendo de la posición de cada elemento dentro del ambiente. Cada elemento determina su propio punto dentro de la formación con referencia a este punto.
- **Referencia con respecto a un líder.** Cada elemento calcula su posición con respecto a un líder, el líder no tratará de mantener su punto de la formación, serán los demás integrantes del grupo responsables de mantenerse dentro de la formación.
- **Referencia con un vecino.** Cada integrante determina su posición de acuerdo a un elemento cercano a él.

4.4.2. Comportamiento pasajes estrechos

Una manera simple de solucionar el problema de los pasajes estrechos, es usar el comportamiento de autoguiado y seleccionar dos nodos como metas, colocar uno de ellos frente a la entrada del pasaje y el otro nodo a la salida del pasaje. La desventaja de esta solución es que el grupo puede amontonarse y crear un conflicto cuando ellos traten de moverse por el pasaje estrecho.

Otra solución sería usando la estrategia siguiendo al líder, así se evitaría el problema del congestionamiento dentro del pasaje, como en la primera solución. En esta estrategia se coloca el grupo frente al pasaje estrecho, se selecciona el agente que se encuentre más cercano a la entrada como líder, se colocan los miembros restantes en una formación columna y estos seguirán al líder a través del pasaje estrecho. La posición que los miembros restantes

toman en la fila depende de la distancia desde el punto en el que se encuentren con respecto a la entrada del pasaje, ésta puede ser formándose del mas cercano al lejano o viceversa. En el caso de formar los miembros restantes desde el mas lejano al cercano, podría tener una desventaja, ya que al hacer la formación de estos podrían tender a amontonarse a la entrada del pasaje.

Uniendo estas ideas, este comportamiento comienza cuando todos los agentes se encuentran bajo un mismo comportamiento u alguna formación, el propósito de esta formación mantener alguna de las formaciones, descritas en la sección anterior, en cualquier etapa de la trayectoria. Existe un disparador que continuamente verifica si los agentes han alcanzado reunirse en el area del pasaje o si están a la entrada de éste. Cuando se encuentran ahí, el rol del agente mas cercano al pasaje se actualiza (como líder). Entonces el líder seguirá la ruta definida para cruzar el pasaje y los restantes seguirán al agente, en una formación en columna.

Algoritmo pasajes estrechos

```
Crear Posición inicial para cada miembro
Dividir el camino en submetas
Guardar en el agente el camino dividido en submetas
Mientras (Agentes no encuentren la meta)
  Si (La meta está en el rango de visión del líder)
    El comportamiento vagar es efectuado por el grupo
  Si_no
    Si (Existe un pasaje estrecho)
      Los agentes toman una formación en columna
      Se realiza el comportamiento siguiendo al líder
    Si_no
      La meta del líder se fija como la submeta siguiente
      Se calculan puntos de la formación para cada
      miembro del grupo
    Fin_si
  Fin_si
Fin_mientras
```

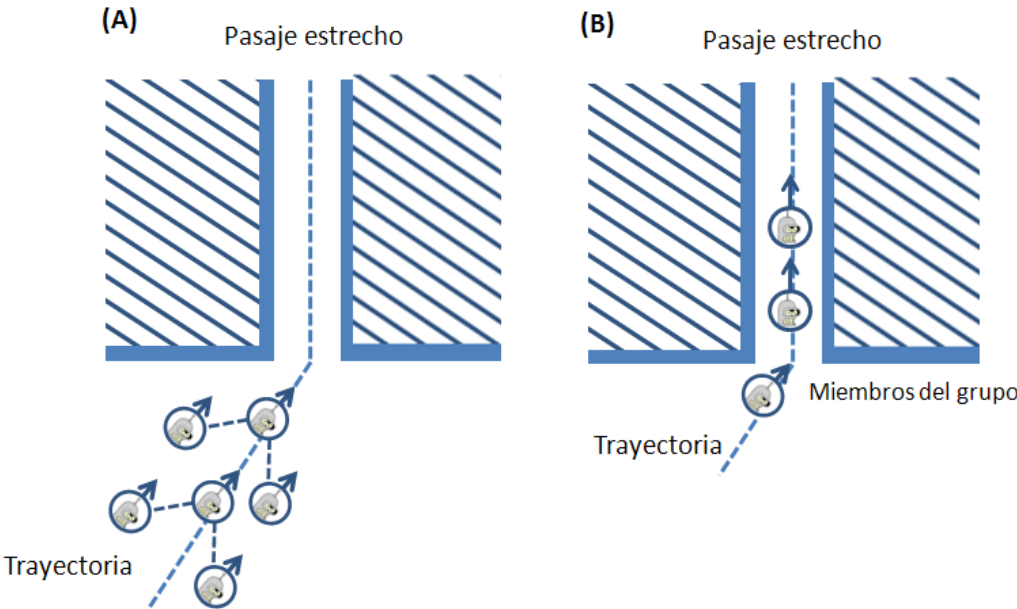


Figura 4.4: Comportamiento de pasajes estrechos. (A) Formación antes del pasaje estrecho. (B) Formación dentro del pasaje estrecho.

4.5. Comportamiento de seguimiento en círculo

El comportamiento de seguimiento puede ser usado por un grupo de caza, que busca o que sigue a otros miembros de grupo a través del ambiente. Por otra parte, podemos tener alguna variante, como el comportamiento de seguimiento en círculo. En este comportamiento los agentes tratan de permanecer en una formación circular alrededor de un agente dado. El propósito de éstos comportamientos es permitir a los agentes proteger o rodear a un agente.

Durante éste comportamiento, los agentes tratan de estar en la misma posición relativa al agente que está rodeado. Los agentes permanecen en círculo y no pierden el ángulo relativo que usan para seguir en su formación en círculo. Como el agente rodeado cambia de direcciones, los agentes que lo siguen modifican la posición de las metas que siguen para que correspondan y puedan tener el mismo ángulo relativo hacia el miembro rodeado.

Algoritmo de seguimiento en círculo

```
Crear Posición inicial para cada miembro
Elegir lider
Guardar la trayectoria general en el lider
Mientras (Líder no encuentre la meta)
  Si (La meta está en el rango de visión del líder)
    El comportamiento vagar es efectuado por todos los miembros
    del grupo
  Si_no
    Para (Cada uno de los restantes miembros del grupo)
      Colocar como submeta el punto del a formación que fué
      calculado con respecto a la siguiente submeta del lider
      Caminar hacia la submeta actual
    Fin_para
  Fin_si
Fin_mientras
```

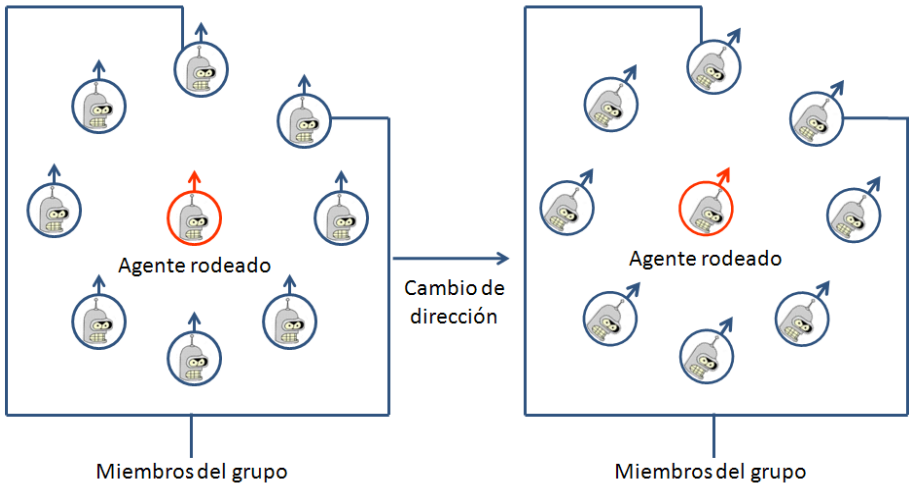


Figura 4.5: Comportamiento de seguimiento en círculo.

4.6. Otros comportamientos

Ahora veamos como la unión de algunos comportamientos nos permiten crear un comportamiento global mas robusto. En esta última parte se realizó un análisis al *comportamiento de pastoreo*. Éste es un tipo de comportamiento grupal, en el que un grupo de personajes (pastores) tratan de controlar el movimiento de otro grupo (rebaño). Este comportamiento puede ser encontrado en muchas formas en la naturaleza y puede tener diversas aplicaciones en la vida cotidiana.

Un pastor es un agente externo que influencia el movimiento de la flota. Una flota es una colección de agentes que se mueven en una manera coordinada [6] mientras reaccionan simultáneamente a factores externos como pastor(es) u otros obstáculos. En este comportamiento el movimiento de un pastor es el movimiento que el agente usa para controlar la rebaño. Algunos comportamientos de pastoreo, incluyendo reunir, cubrir, patrullar y recolectar, pueden ser generados usando movimientos comunes.

Se definen dos formaciones para pastores, una formación en línea y una formación en arco. La formación en línea coloca los posiciones en un segmento de línea colocada detrás de la flota. Esta línea es perpendicular a la línea entre el centro de la flota y el punto medio de la línea formada. Una formación en línea es útil mientras se está empujando la flota hacia la submeta. Una formación en arco es un segmento semicircular detrás de la flota, centrado con respecto al centro de la flota y perpendicular a la submeta.

Al inicio del comportamiento, el grupo de personajes que tienen el papel de rebaño, pueden estar dispersos o agrupados dentro del ambiente virtual, si el grupo está agrupado significa que todo el rebaño se encuentra reunido dentro de un círculo, donde el centro es el punto inicial de la trayectoria, en el caso de que se encuentre dispersos los otros actores que juegan el papel de pastores tendrán que adoptar un comportamiento de recolección para reunir a cada una de las ovejas en un mismo punto conocido como punto inicial.

Los pastores se posicionan en un punto detrás del rebaño, adoptando alguna de las formaciones mencionadas anteriormente, el punto medio del grupo de ovejas es tomado como referencia para trazar un círculo, que indicará la zona donde las ovejas deben permanecer, la submeta siguiente es el punto siguiente dentro la trayectoria, y los pastores se moverán de manera que la flota no salga del círculo, y poder encontrar la meta, en este caso si no se ha llegado a la meta final, la siguiente meta es el siguiente punto de la trayectoria. Antes de llegar a la meta los pastores deben verificar que el rebaño está reunido dentro del círculo.

Capítulo 5

Herramienta RMP3D

5.1. Arquitectura de RMP3D

En este trabajo de tesis, nos interesamos en realizar comportamientos complejos partiendo de comportamientos simples, estos comportamientos son conocidos como comportamientos de Reynolds y fueron descritos anteriormente, por otro lado los comportamientos complejos desarrollados utilizan métodos roadmap, que pueden resolver rápidamente diversas consultas de planificación de caminos.

Para realizar esto, nos hemos basado en una herramienta dedicada a la planificación de movimientos, la herramienta se denomina RMP3D¹, dicha herramienta sirvió de base para la implementación de nuestro esquema de animación de caracteres autónomos.

En la Figura 4.1 se muestra la arquitectura del sistema que está compuesta por los siguientes módulos:

- **Modelado.** En este módulo se almacenan y manipulan los datos del personaje virtual, los datos del ambiente virtual, configuración inicial y final.
- **Algoritmos de planificación de movimientos.** En este módulo se construye un roadmap mediante técnicas probabilísticas PRM. Los métodos de muestreo implementados son los siguientes:
 - **Muestreo uniforme.** Se crean las muestras tomando valores de posición y orientación aleatoriamente en el CS_{libre} . Se consideran hasta seis grados de libertad.
Se agregaron además, un grupo de métodos deterministas, los

¹Esta herramienta se desarrolló en la FCC-BUAP

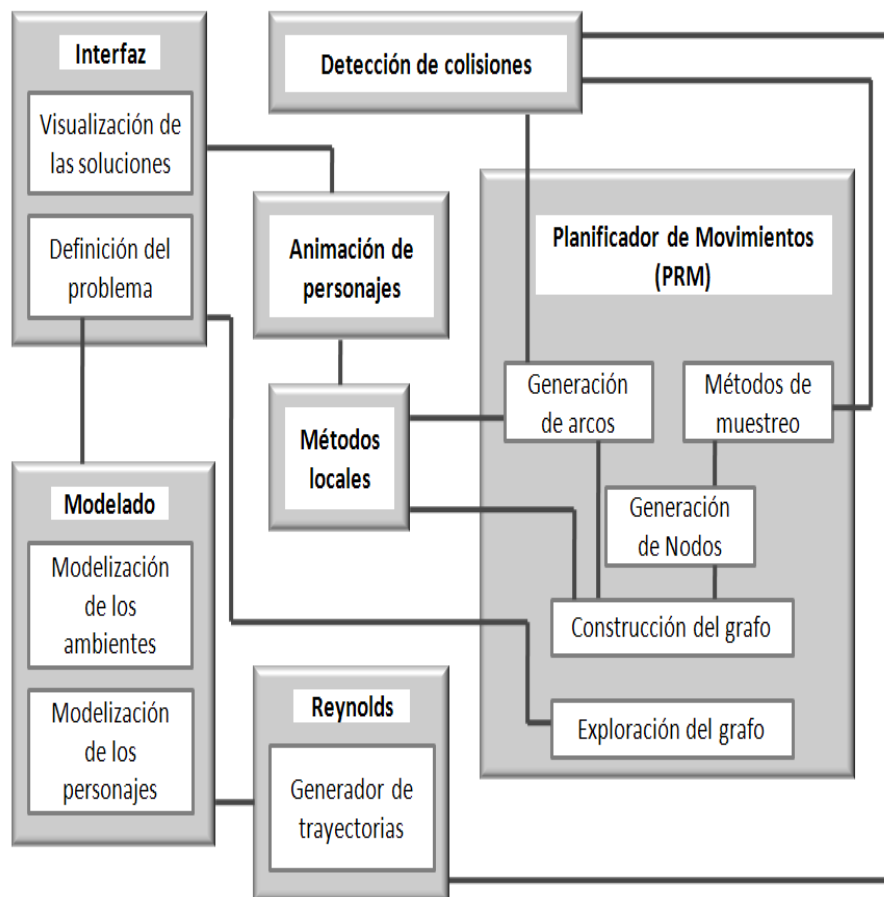


Figura 5.1: Arquitectura de RMP3D.

cuales nos ofrecen una mejor cobertura del espacio libre, estos métodos son los siguientes:

- Series de Halton,
- Series de Faure,
- Series de Sobol,
- Series de Niederreiter,
- **Gaussiano.** Las muestras que se agregan al grafo son las más cercanas a los obstáculos, se mejora la cobertura de las regiones difíciles del espacio libre.
- **Vertex.** Sea V el conjunto de todos los vértices que componen el espacio de configuraciones, las muestras son generadas a una distancia k cerca de v con $v \in V$

Los **métodos locales** implementados para la construcción del grafo son los siguientes:

- **Método lineal.** Calcula un segmento de línea entre dos configuraciones, este método trabaja adecuadamente para cualquier sistema holonómico (por ejemplo, un brazo manipulador, o un objeto de vuelo libre).
- **Método de solo translación.** Este método solo calcula movimientos de solo translación, es de utilidad para probar las distintas variantes de PRM en escenarios que contienen pasajes estrechos.
- **Método de campo potencial.** Está basado en el cálculo de técnicas de gradiente descendiente y proporciona resultados satisfactorios, el único problema es que consume mucho tiempo de cómputo. La meta de este módulo es encontrar el camino entre dos configuraciones dadas, sin considerar los obstáculos.
- **Detector de colisiones.** El algoritmo integrado es PQP² para determinar si una configuración dada esta libre de colisión o no (esto se determina haciendo múltiples llamadas al algoritmo de la detección de interferencias y cálculos de distancia entre dos objetos). PQP es una librería que ejecuta tres tipos diferentes de consultas de proximidad en un par de modelos geométricos integrados por triángulos.
- **Animación de caracteres.** En este módulo se realiza la animación recibiendo como parámetros una trayectoria libre de colisión y el número de caracteres. Se describirá mas adelante con más detalle.

²Un paquete de detección de colisiones desarrollado en la Universidad de Carolina del Norte en Chapel Hill

- **Módulo de Reynolds.** Este módulo es el encargado de generar la animación para los comportamientos Reynolds descritos anteriormente, hace uso del detector de colisiones, además de utilizar como parámetro el número de personajes.
- **Interfaz.** Este módulo es el encargado de mostrar los resultado en pantalla, así como permitirle al usuario especificar los parámetros para PRM, como el método de muestreo, el método local, la forma en que debe construirse el roadmap: tiempo de construcción, número de nodos o número de nodos conectados en el grafo, etc.

La siguiente subsección presenta una parte del diseño de la herramienta Crowd-BUAP. Dicha herramienta es una especialización de la herramienta RMP3D que se dedica a la animación de personajes autónomos.

5.1.1. Diseño de la herramienta Crowd-Buap

En este caso sólo haremos énfasis en los diagramas de casos de uso y los diagramas de clases.

Las siguientes figuras muestran los casos de uso.

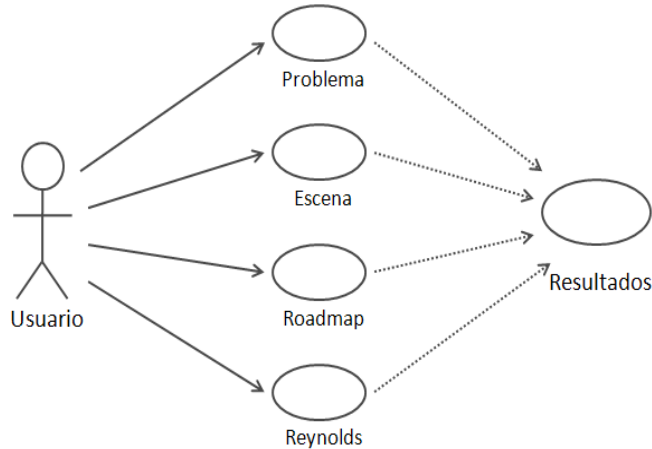


Figura 5.2: Caso de uso de la herramienta Crowd-Buap

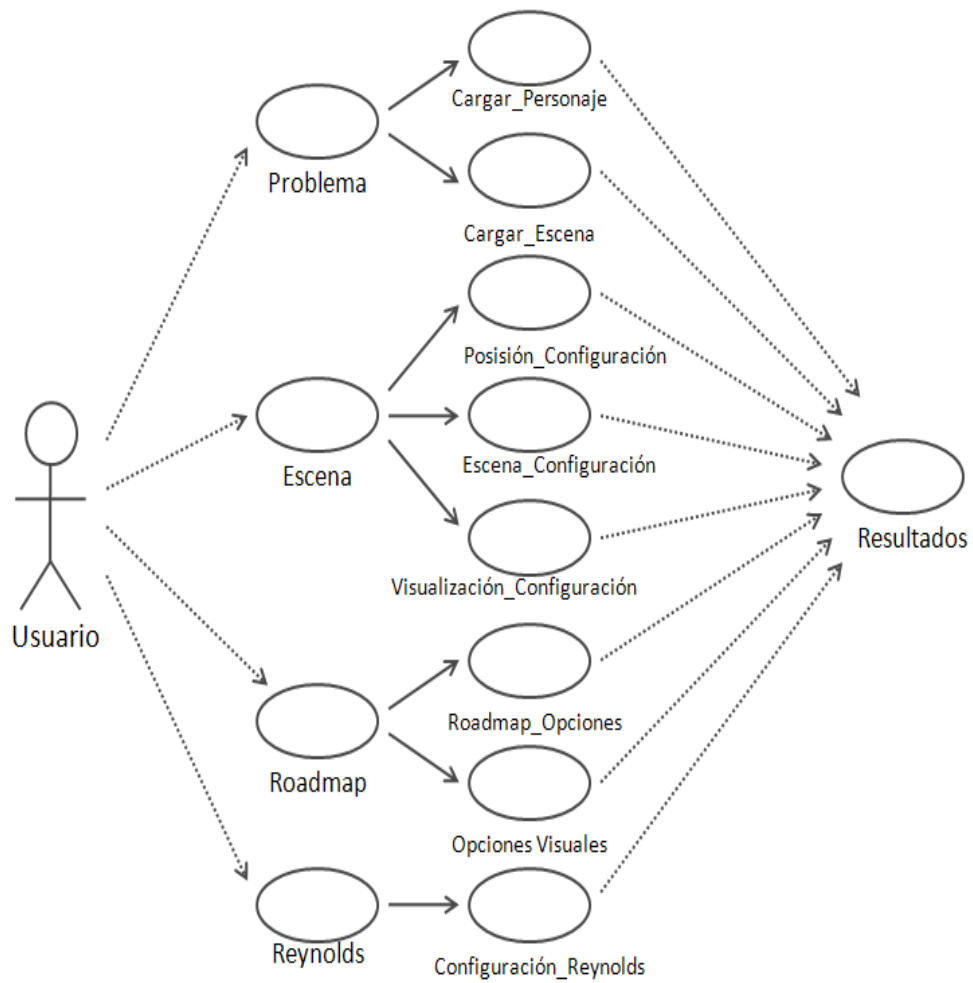


Figura 5.3: Subcaso de uso de la herramienta Crowd-Buap

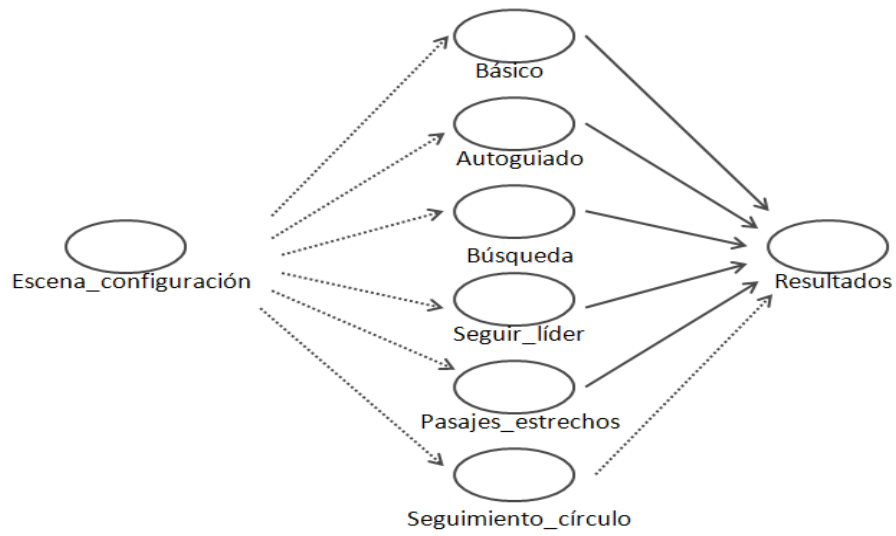


Figura 5.4: Subcaso de uso de la herramienta Crowd-Buap

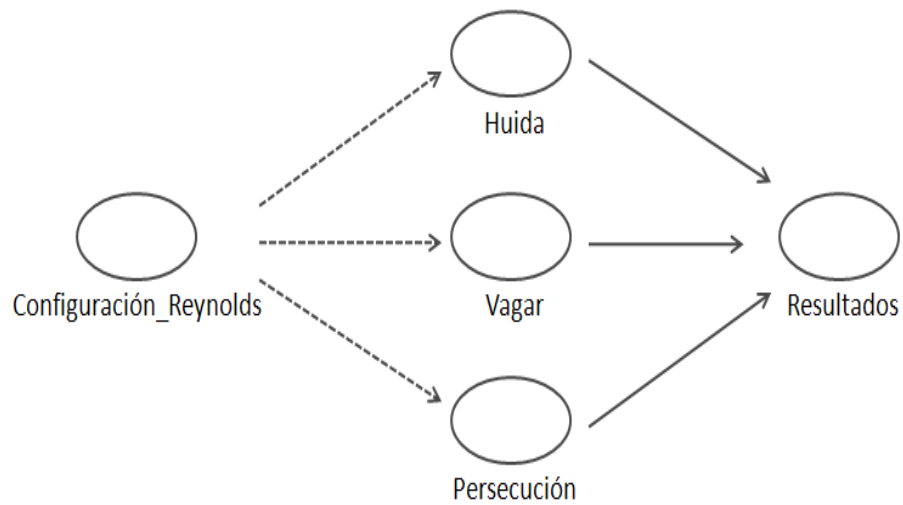


Figura 5.5: Subcaso de uso de la herramienta Crowd-Buap

Las siguientes figuras muestran los diagramas de clases de la herramienta.

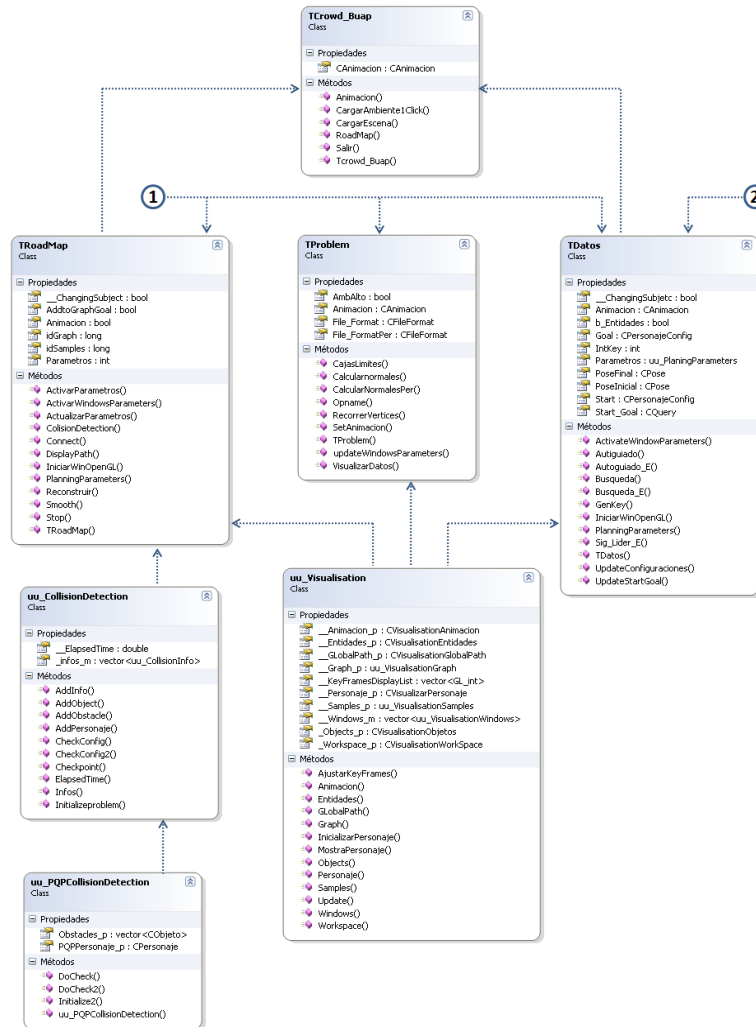


Figura 5.6: Diagrama de clases de la herramienta Crowd-Buap

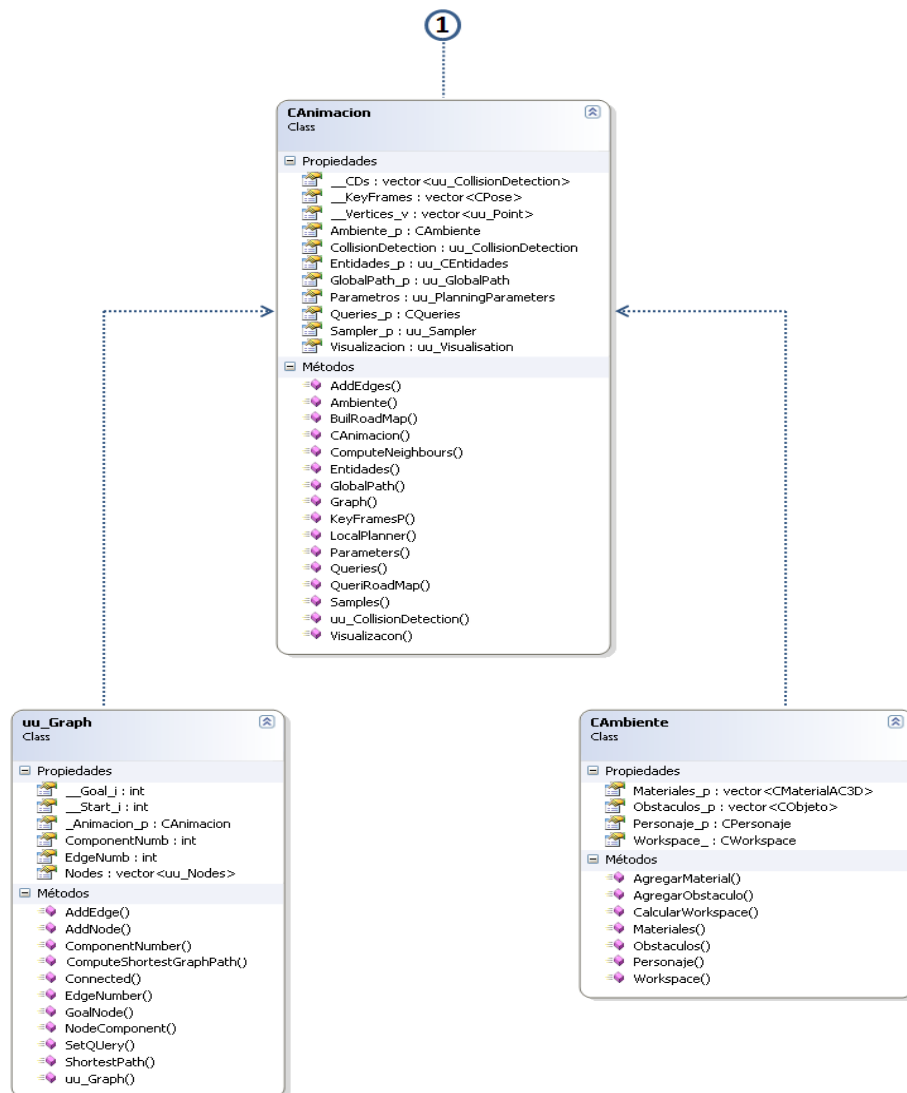


Figura 5.7: Continuación del diagrama de clases de la herramienta Crowd-Buap

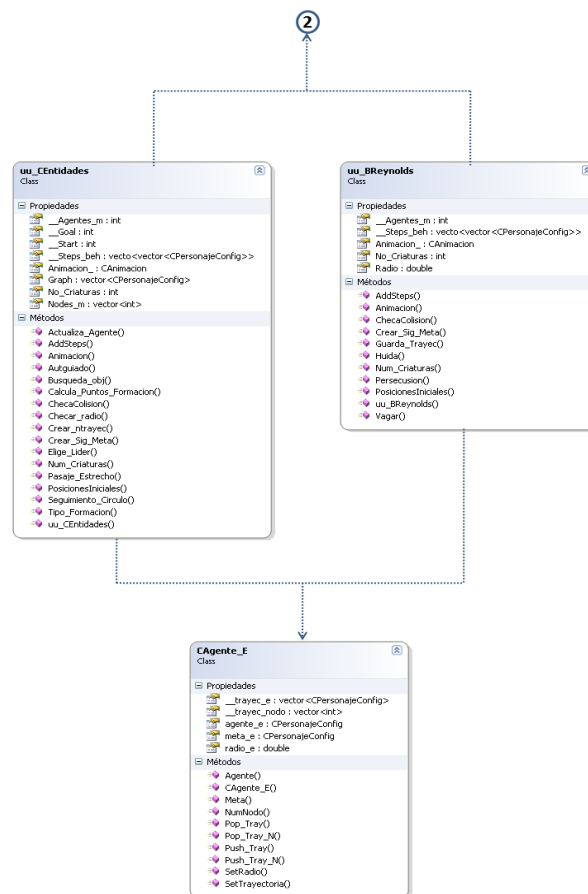


Figura 5.8: Continuación del diagrama de clases de la herramienta Crowd-Buap

5.2. Módulo de Reynolds

El objetivo de este módulo es generar la animación de comportamientos básicos de Reynolds, los comportamientos implementados son los descritos en el capítulo 2:

- Comportamiento de Llegada.
- Comportamiento de Huida.
- Comportamiento de Vagar.
- Comportamiento de Persecución Simple.

Los componentes principales que contiene este módulo son: un ambiente y uno o más agentes. Cada agente desempeña un papel que depende del comportamiento que éste adopte. La interacción del personaje con el ambiente es apoyada con el módulo que contiene el detector de colisiones.

Reynolds describe un conjunto de comportamientos simples que sirven como base para la generación de comportamientos más complejos, éstos comportamientos utilizan alguna técnica PRM para navegar a través de un ambiente complejo con obstáculos.

La combinación de éstos comportamientos simples se realiza dentro del módulo de animación de caracteres para lograr la simulación de comportamientos complejos que se asemejan más a la realidad.

5.3. Módulo de animación de personajes

Este módulo tiene como objetivo la integración de técnicas de agrupamiento a esta herramienta, es decir, acoplamos reglas de comportamiento en miembros individuales, en nodos y en arcos del roadmap.

El trabajo de Parker apoya el uso global de la información, para hacer o desarrollar comportamientos de grupos sofisticados [15]. En particular, concluyó que el conocimiento global debería ser usado para proporcionar una guía general para que el conocimiento local influya en acciones a largo plazo. También sugirió que la información local debería ser usada para asentar conocimiento global en la situación actual.

Los componentes generales de este módulo son uno o más agentes y un ambiente. Cada agente es miembro del grupo global y tiene distintos atributos asociados con el grupo, como son: posición, meta y trayectoria. El

ambiente almacena información global acerca del roadmap, la cual es compartida por todos los agentes. Los nodos del roadmap almacenan atributos del nodo como: la posición, los arcos y el número de arcos, y tiene funciones para la navegación como: regresar el mínimo peso.

Los comportamientos que se implementaron con estas técnicas están descritos en el capítulo 3 y son los siguientes:

- Comportamiento de Autoguiado.
- Comportamiento de Búsqueda de un objetivo.
- Comportamiento de Siguiendo al líder.
- Comportamiento de Pasajes estrechos.
- Comportamiento de Seguimiento en círculo.

Capítulo 6

Resultados Experimentales

6.1. Entorno de trabajo

En este capítulo se muestra una breve descripción de la herramienta desarrollada al igual que algunos resultados experimentales.

El software está desarrollado con el compilador C++ Builder V.5.0. La base de esta aplicación es una herramienta que contenía métodos de planificación probabilísticos, un detector de colisiones, una interfaz gráfica que permite mostrar los resultados al usuario y además algunos comportamientos grupales. Esta aplicación se ha modificado de tal manera que se adaptaron algunos de los comportamientos descritos en capítulos anteriores. El nombre de la herramienta construida es Crowd-BUAP 2.0.

La interfaz gráfica está desarrollada con OpenGL y Forms. Ésta interfaz permite al usuario definir el problema de planificación de movimientos y algunas opciones de los comportamientos Reynolds, así como también el usuario puede visualizar cada etapa de la planificación, modificar parámetros por ejemplo el método local, número de nodos, etcétera, además de poder seleccionar alguno de los comportamientos desarrollados.

El ambiente 3D y los personajes están diseñados con la ayuda de Invis AC3D¹. Se requiere que estén compuestos por solo triángulos, ya que es un requisito obligatorio para el detector de colisiones PQP. En nuestro caso los obstáculos son considerados como estáticos.

Las siguientes figuras muestran algunas de funciones de la herramienta Crowd-BUAP.

¹una herramienta de modelado en 3D

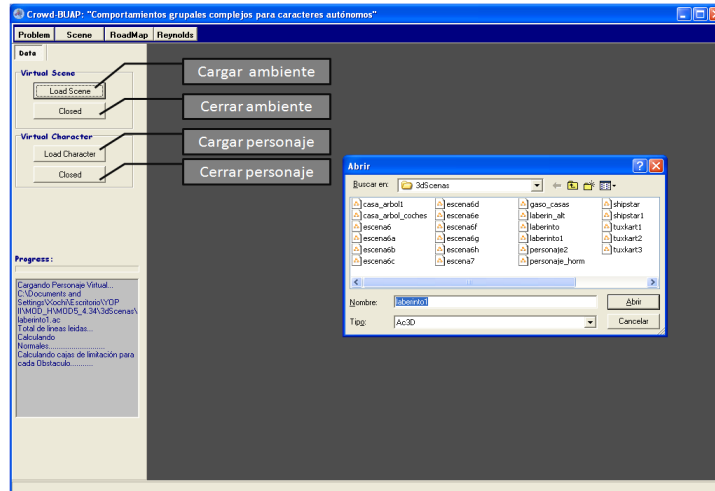


Figura 6.1: Selección de personaje y de la escena.

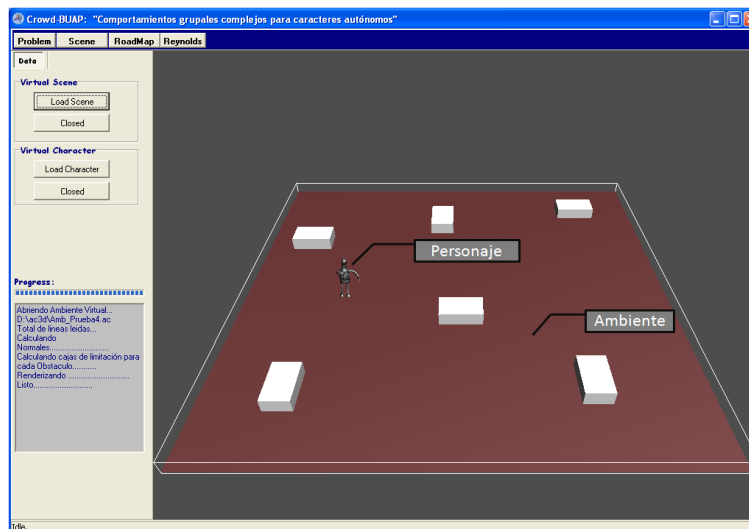


Figura 6.2: El personaje autónomo y su ambiente.

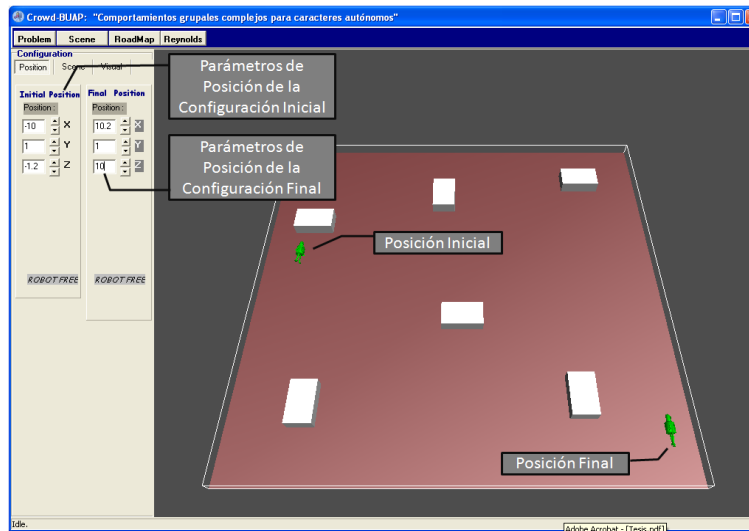


Figura 6.3: Posición inicial y final del personaje.

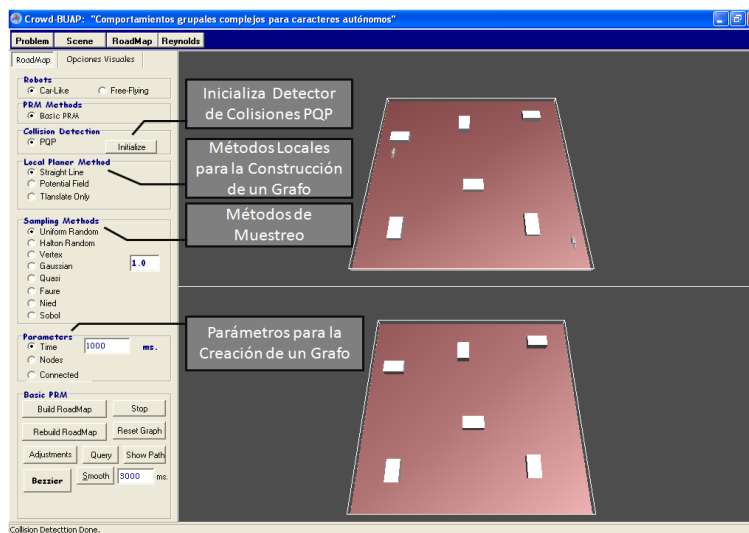


Figura 6.4: Parámetros del Roadmap.

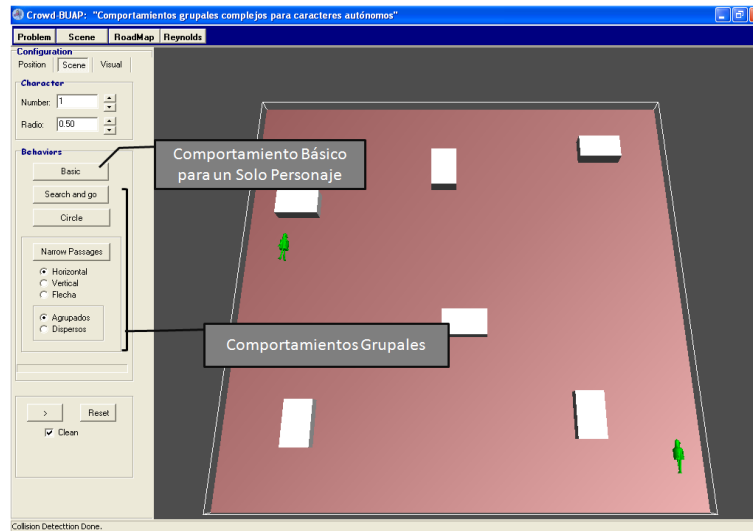


Figura 6.5: Comportamientos grupales y comportamiento básico disponibles en la herramienta.

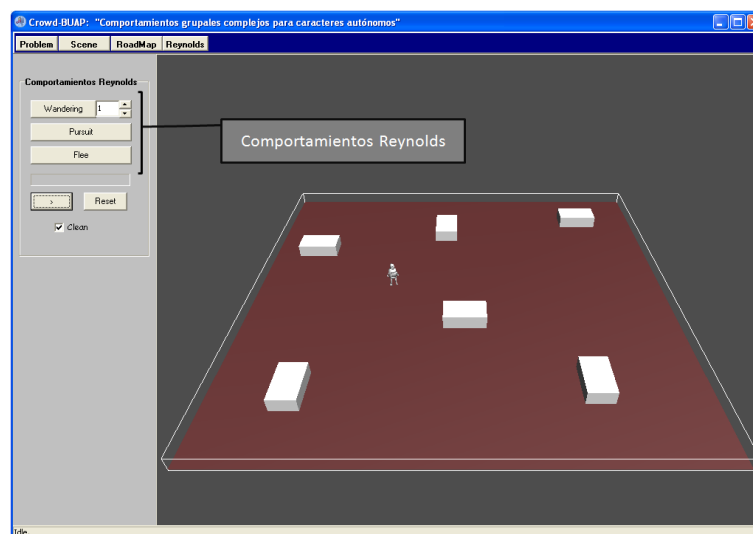


Figura 6.6: Comportamientos Reynolds.

6.2. Resultados

Todos las pruebas fueron realizadas en una máquina Intel Pentium D 820 con procesador de 2.8 GHz con memoria de 3 GB. Para comenzar a presentar los resultados iniciaremos mostrando algunas imagenes de los comportamientos Reynolds, como ya mencionamos anteriormente éstos comportamientos y los comportamientos grupales fueron implementados en la herramienta Crowd-buap 2.0.

6.2.1. Comportamientos Reynolds

En la Figura 5.7 se muestra un ambiente que simula a una ciudad virtual, éste contiene algunos edificios y calles en donde se puede transitar. En esta ciudad virtual se encuentra un grupo de 7 personajes virtuales, cada personaje tiene como objetivo recorrer el ambiente bajo el comportamiento vagar, es decir, que cada personaje recorrerá la ciudad de manera aleatoria sin colisionar con el ambiente ó con los demás personajes.

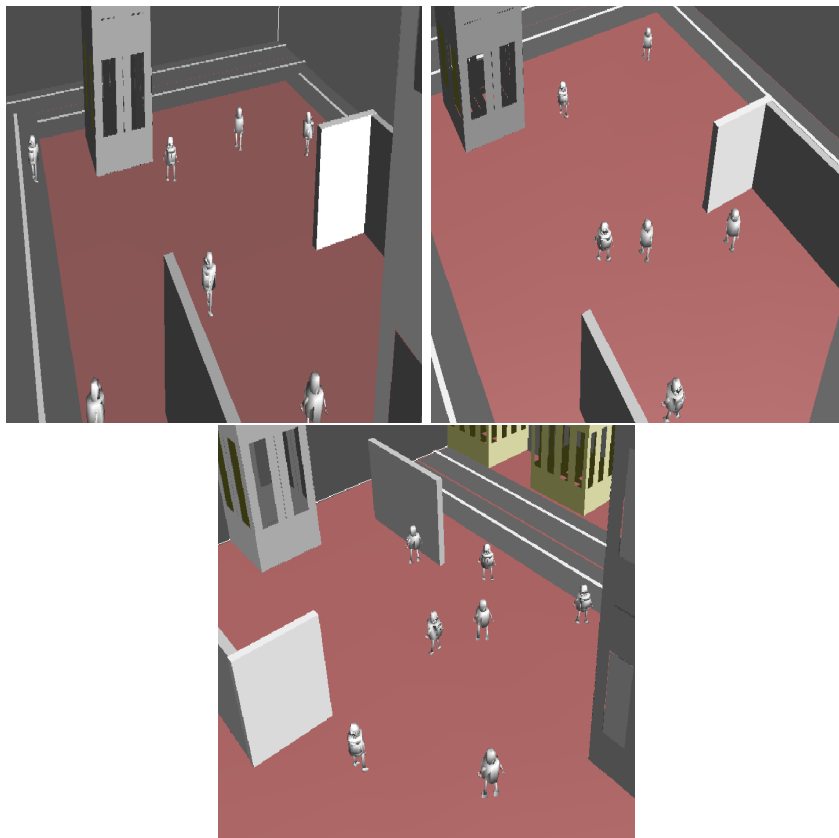


Figura 6.7: Ejemplo del comportamiento vagar.

En la fFigura 5.8 se presentan dos personajes dentro de un ambiente que simula un valle de rocas, en donde se puede recorrer este valle a través de ellas. Para recorrer el ambiente, los personajes ejecutan el comportamiento de persecución, como se puede observar figura, uno de ellos persigue al otro.

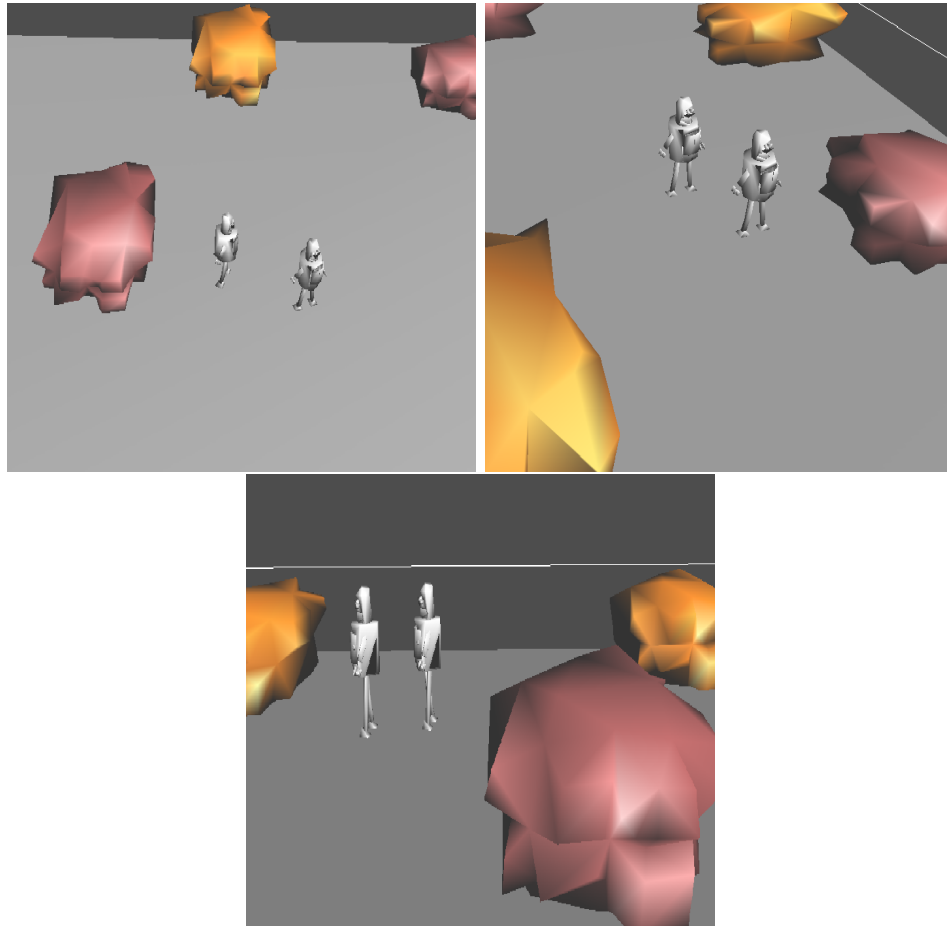


Figura 6.8: Ejemplo del comportamiento de persecución.

En la siguiente prueba se ejemplifica el comportamiento de huida, éste comportamiento es similar al de persecución, mostrado en la figura anterior, la parte interesante de éste comportamiento es que uno de los personajes tiene la capacidad de determinar a que distancia se encuentra el otro personaje para decidir en que momento de huir de él. A ésta distancia se le llama distancia de pánico. La Figura 5.9 muestra éste comportamiento.

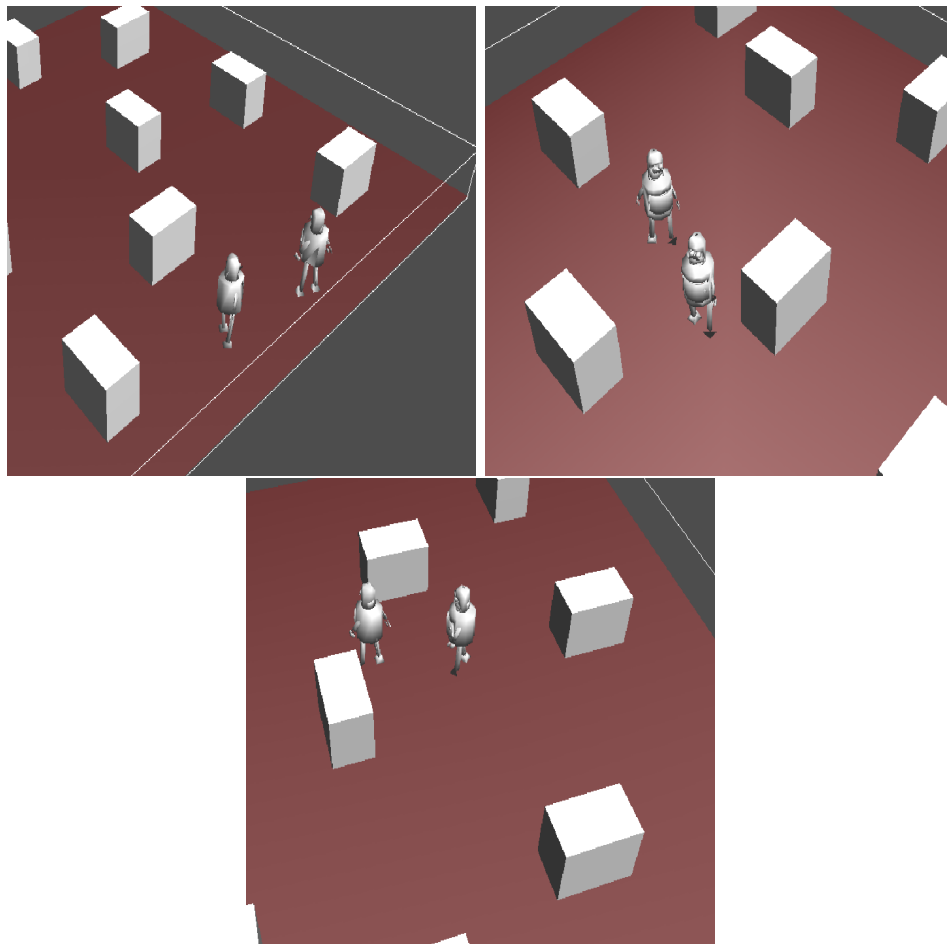


Figura 6.9: Ejemplo del comportamiento de huída.

6.2.2. Comportamiento Pasajes Estrechos

En esta sección se mostrarán los resultados obtenidos por el comportamiento de pasajes estrechos. En la Figura 5.10 se ejemplifican dos diferentes ambientes virtuales, el primero simula ser un parque que contiene dos estacionamientos, uno para motos y otro para autos, en el cual, se puede llegar al otro extremo solo si se cruza el corredor central. El otro ambiente virtual simula ser un restaurante de un parque acuático que consta de tres secciones, para poder acceder a cada una de ellas, es necesario cruzar a través de los puentes que las unen.

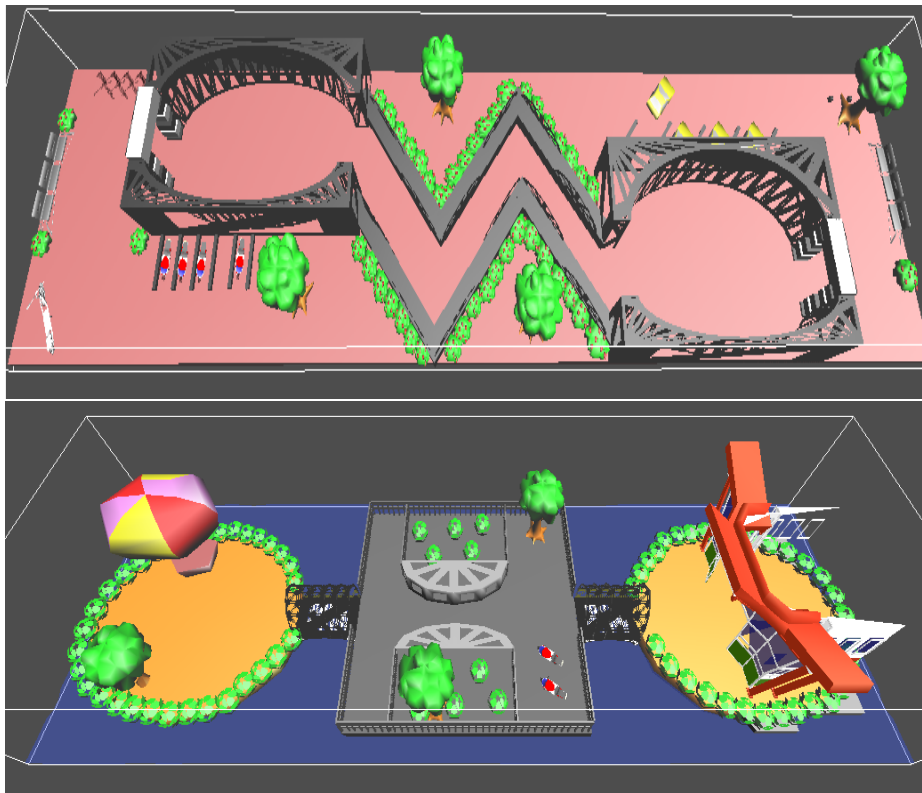


Figura 6.10: Ambientes virtuales.

El objetivo de los personajes virtuales en ambos ambientes es viajar de un extremo a otro. En el primero, los personajes se encuentran en el estacionamiento de motos y su meta es llegar al estacionamiento de autos. En el parque acuático, la meta de los personajes es llegar a la isla donde se encuentra el globo aerostático, teniendo como punto de partida el edificio. Para lograr estos dos objetivos, en ambos ambientes, los personajes tendrán que poner en marcha el comportamiento de pasajes estrechos.

Durante la trayectoria los personajes mantienen alguna formación, como las descritas en el capítulo 3, estas formaciones son: formación en línea, columna, y flecha, que se muestran en la Figura 5.11.

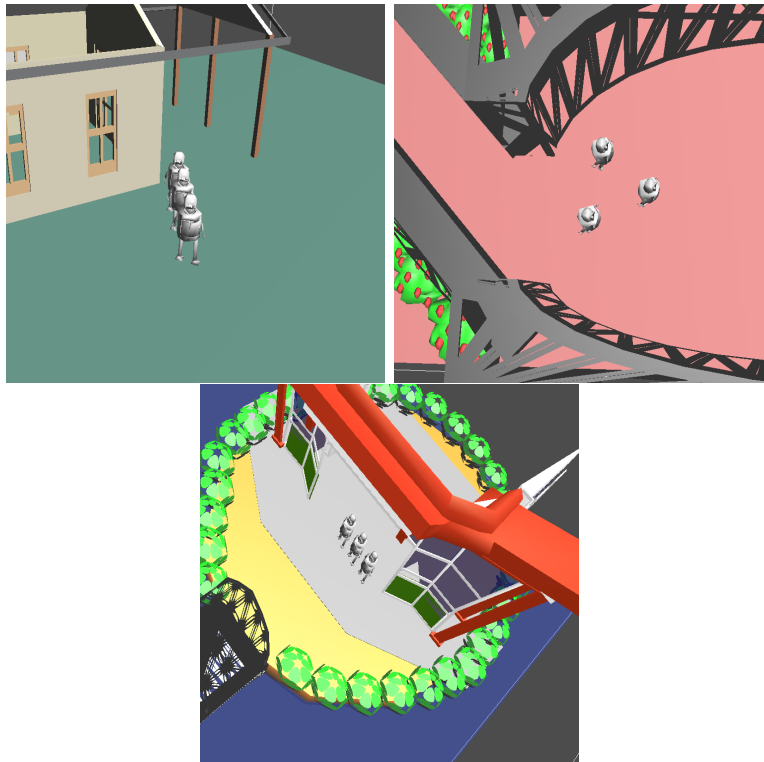


Figura 6.11: Formaciones columna, flecha, línea.

Como ya mencionamos, el comportamiento de pasajes estrechos tiene la característica que durante toda la trayectoria, el grupo de personajes virtuales mantienen una formación específica, sin embargo, en el caso de que durante la trayectoria se encontraran algún pasaje estrecho donde la formación de los personajes no pueda pasar, el grupo debe cambiar a otra formación que les permita atravesar el pasaje estecho sin problema.

En la Figura 5.12 se muestra como los personajes cambian de formación debido a que durante su trayecto se encuentra algún pasaje estrecho. En la Figura 5.12 (A) se muestra el ambiente virtual que simula el parque con estacionamientos, cuando los personajes se encuentran al inicio del corredor listos para llegar al otro extremo, se ven en la necesidad de cambiar su formación para seguir adelante. Del mismo modo en la Figura 5.12 (B) se observa que los personajes se encuentran antes de cruzar el primer puente, lo que les implica cambiar su formación para poder llegar a la segunda sección

del ambiente virtual del parque acuático.

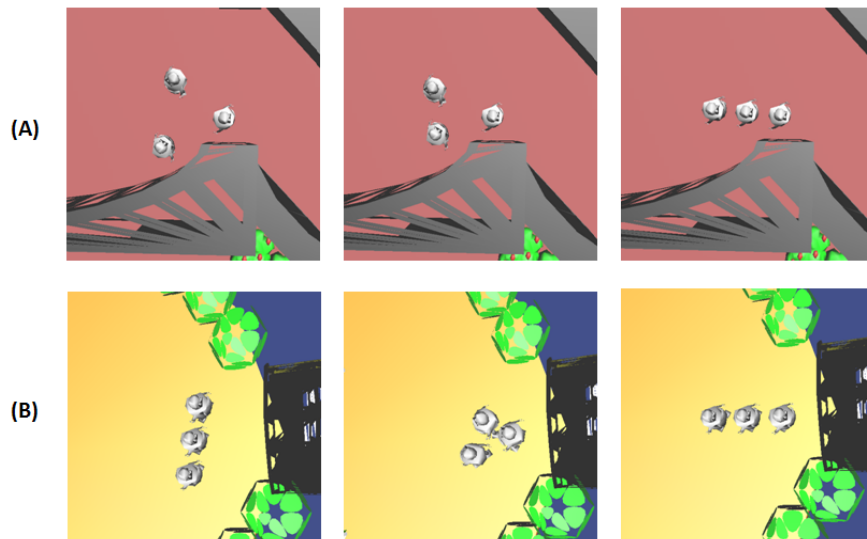


Figura 6.12: Cambio de la formación antes de entrar al pasaje estrecho.

Una vez que los personajes han entrado al pasaje estrecho, deben mantener la formación en columna hasta que salgan de él, como se puede observar en la Figura 5.13 (A) y (B) los personajes se encuentran dentro del corredor y el puente respectivamente, con la formación antes mencionada.

Después que los personajes salen del pasaje estrecho deben regresar a la formación que llevaban antes de entrar al pasaje estrecho. En la Figura 5.14 podemos observar como se realiza el cambio de formación, en la Figura 5.14 (A) los personajes salen del corredor y cambian su formación para poder seguir con la trayectoria. Para el ambiente del parque acuático, los personajes salen del puente, se forman en línea y continúan con su trayectoria, como se muestra en la Figura 5.14 (B).

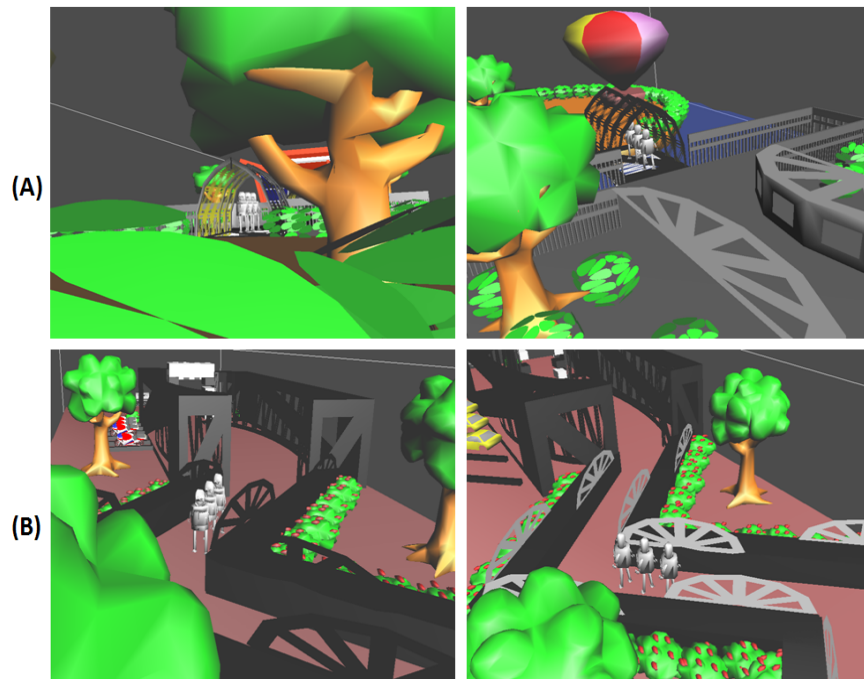


Figura 6.13: Personajes dentro de un pasaje estrecho.

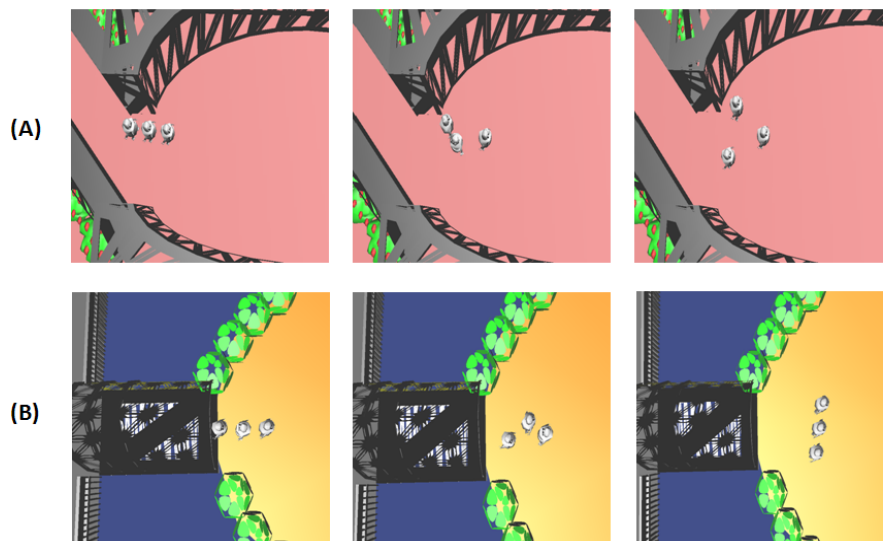


Figura 6.14: Cambio de formación después de cruzar el pasaje estrecho.

Como pudimos observar en las imágenes mostradas anteriormente, tenemos una formación que podemos considerar como auxiliar para la realización de éste comportamiento, ésta formación es la *formación en columna*, ya que este tipo de formación nos permite atravesar pasajes estrechos sin ninguna complicación. Además, esta formación puede ser utilizada para navegar un ambiente que contenga o no pasajes estrechos.

Para mostrar esta formación diseñamos un ambiente virtual que representa una casa que contiene en su interior dos patios, la Figura 5.15 presenta el ambiente diseñado, donde los personajes tienen que viajar desde uno de ellos hacia el otro.

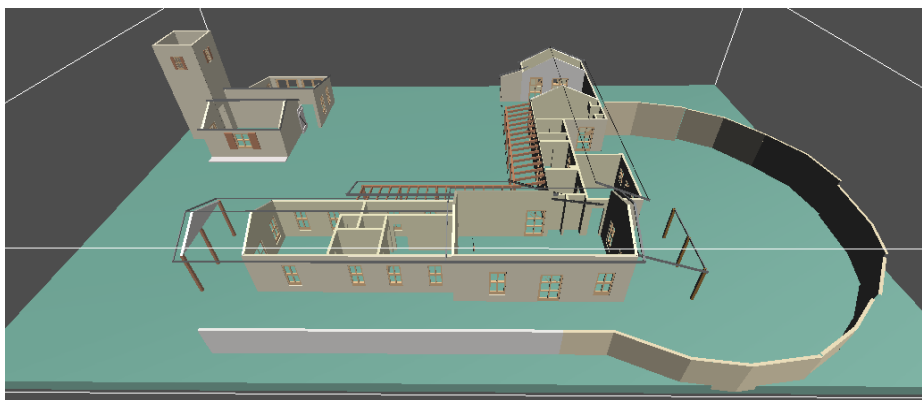


Figura 6.15: Ambiente virtual.

La Figura 5.16 muestra como los personajes viajan a lo largo del escenario usando ésta formación.



Figura 6.16: Formación en columna.

Una vez que los personajes llegan a su destino toman un comportamiento de vagar descrito anteriormente, como se muestra en la Figura 5.17 incisos (A), (B), (C) respectivamente, en el cual, tiene como finalidad recorrer el ambiente de manera aleatoria. Es decir, en el ambiente del estacionamiento

los personajes vagan en el estacionamiento de coches, en el ambiente del parque acuático los personajes vagan en la sección del globo aerostático y en el ambiente del edificio los personajes vagan en el patio que es su objetivo final.

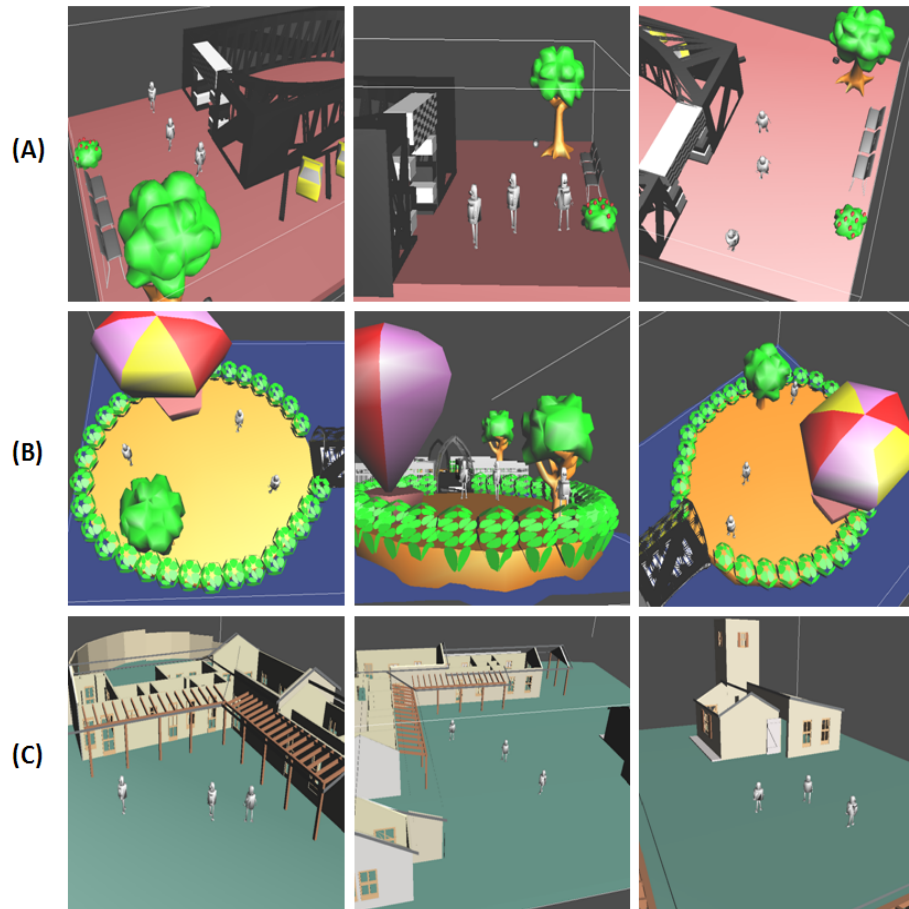


Figura 6.17: Todos los personajes vagan dentro de su respectivo ambiente virtual.

Capítulo 7

Conclusiones y trabajos futuros

El objetivo principal de la elaboración de este trabajo de tesis, fué integrar un conjunto de comportamientos simples (comportamientos de Reynolds), para la animación de personajes autónomos que se desenvuelven dentro de una ambiente virtual simple, en la herramienta Crowd-Buap.

Sin embargo, el alcance de los comportamientos simples es limitado, por ello una aportación importante es la unión de éstos comportamientos, con algunos comportamientos grupales ya existentes dentro de la herramienta, además de utilizar técnicas PRM que también ya estaban contenidas dentro de ésta.

El comportamiento de pasajes estrechos desarrollado, es el resultado de la unión de tres conceptos importantes que son: comportamientos simples, comportamientos grupales y además formaciones. Al desarrollar éste comportamiento se incluyó además de los comportamientos simples y grupales existentes, el concepto de formaciones, que surgió de la necesidad de mantener al grupo de personajes de una manera ordenada dentro del pasaje estrecho. Ésto nos lleva a la idea, de que un comportamiento complejo se puede elaborar de manera sencilla haciendo la integración de uno o más comportamientos simples. Ésta combinación hace aún mas complejo el resultado obtenido permitiendo lograr un comportamiento más realista.

Las formaciones además pueden tener otras aplicaciones ya que pueden ser de utilidad en otras situaciones, por ejemplo búsqueda y rescate, tareas de agricultura o patrullaje de seguridad.

Además del comportamiento desarrollado, el análisis de un comportamiento mucho más complejo, como lo es el comportamiento de pastoreo nos da una perspectiva del alcance que puede tener la combinación de comportamientos simples, ésta combinación puede tener diversas aplicaciones dentro de muchas areas de la investigación. Simular el comportamiento coordinado de un grupo de pastores y otro de ovejas, ha sido estudiado en diferentes campos incluyendo la robotica, animación por computadora y juegos. Éste comportamiento en general tiene aplicaciones más allá de construir robots que pastoreen un grupo de animales. Por ejemplo, los comportamientos de pastoreo pueden ser utilizados para construir robots que trabajen cooperativamente para recolectar aceite derramado en el mar por un barco petrolero, similarmente también pueden ser utilizados para mantener a personas fuera de un área restringida o de peligro, o servir como guías de turistas en un museo.

Bibliografía

- [1] Agha G. *Actors: a model of concurrent computation in distributed systems*. Cambridge, MA: MIT Press, 1986.
- [2] Bell S. *Pavlovian polygons*. Proceedings of Computer Graphics, 1990:253-62.
- [3] Blumberg BM, Galyean A. *Multi-level direction of autonomous creatures for real-time virtual environments*. Proceedings of Computer Graphics, 1995:47-54.
- [4] Burtnyk N, Wein M. *Computer generated key-frame animation*. Journal of Society Motion Picture Television Engineers 1971;80(3):149-53.
- [5] Cliff Dave and Miller Geoffrey. *Co-Evolution of Pursuit and Evasion II: Simulation Methods and Results*, From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96), Maes, Mataric, Meyer, Pollack, and Wilson editors, ISBN 0-262-63178-4, MIT Press,1996.
- [6] C. W. Reynolds. *Flocks, herds, and schools: A distributed behavioral model*. In Computer Graphics,1987; 25,34.
- [7] Haumann DR. *Animating using behavioral simulation*. Proceedings of NCGA, 1988:672-80.
- [8] Haumann DR, Parent RE. *The behavioural test bed: obtaining complex behaviour from simple rules*. Visual Computer 1988;4(6):332-47.
- [9] Hernandez Velazquez X., *Animación de personajes autónomos utilizando técnicas PRM*. Tesis de licenciatura, FCC-BUAP, 2006;
- [10] Hubbard Philip. *Approximating Polyhedra with Spheres for Time-Critical Collision Detection*, ACM Transactions on Graphics, 15(03), pages 179-210, 1996.

-
- [11] Isaacs Rufus. *Differential Games: A Mathematical Theory with Application to Warfare and Pursuit*, Control and Optimization, John Wiley and Sons, New York, 1965
- [12] Latombe, J. C. *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.
- [13] Mah S, Calvert T, Havens W. *NSAIL PLAN: An experience with constraint-based reasoning in planning animation*. Proceedings of Computer Animation 94, 1994:83-92.
- [14] Noser H, Renault O, Thalmann D. *Navigation for digital actors based on synthetic vision, memory, and learning*. Computers and Graphics 1995;19(1):7-19.
- [15] Parker L. E. *Designing control laws for cooperative agent teams*. In IEEE International Conference on Robotics and Automation, 582-587, 1993.
- [16] Pugh J. *Actors the stage is set*. ACM SIGPLAN NOTICES, 1984;19:(Part 3) 61-5.
- [17] Reich BD, Ko H, Becket W, Badler NI. *Terrain reasoning for human locomotion*. Proceedings of Computer Animation 94, 1994:76-82.
- [18] Renault O, Magnenat-Thalmann N, Thalmann D. *A vision based approach to behavioural animation*. Journal of Visualization and Computer Animation 1990;1(1):18-21.
- [19] Reynolds CW. *Flocks, herds, schools: a distributed behavioural model*. Proceedings of the SIGGRAPH 87 Computer Graphics 1987;21(4):25-34.
- [20] Ridsdale G. *Connectionist modelling of skill dynamics*. Journal of Visualisation and Computer Animation, Part 2 1990;1(2):66-72.
- [21] Takeuchi T, Unuma M, Amakawa K. *Path planning and its application to human animation systems*. Proceedings of Computer Animation 92, 1992:163-74.
- [22] Thalmann D. *Robotics methods for task-level and behavioural animation*. Scientific Visualisation and Graphics Simulation 1990:129-47.
- [23] Tucker Balch, Ronald C. *Behaviour-based Formation Control for Multi-robot Teams*. IEEE Transactions on Robotics and Automation, 1999.
- [24] Tu X, Terzopoulos D. *Artificial fishes: physics, locomotion, perception, behaviour*. Proceedings of Computer Graphics, 1994:43-50.