

Comportamientos de dirección para personajes autónomos

Craig W. Reynolds

Sony Computer Entertainment América

919 East Hillsdale Boulevard

Foster City, California 94404

craig_reynolds@playstation.sony.com

<http://www.red.com/cwr/>

cwr@red.com

Palabras clave: Técnicas de animación, entornos virtuales/interactivos, juegos, simulación, animación conductual, agente autónomo, situado, encarnado, reactivo, vehículo, dirección, planificación de trayectoria, seguimiento de trayectoria, persecución, evasión, evitación de obstáculos, evitación de colisiones, agrupamiento, comportamiento grupal, navegación, vida artificial, improvisación.

Abstracto

Este artículo presenta soluciones para un requisito de los personajes autónomos en la animación y los juegos: la capacidad de navegar por su mundo de una manera realista e improvisada.

Estos “comportamientos de dirección” son en gran medida independientes de los detalles de los medios de locomoción del personaje. Se pueden utilizar combinaciones de comportamientos de dirección para lograr objetivos de nivel superior (por ejemplo: ir de aquí a allá evitando obstáculos, seguir este corredor, unirse a ese grupo de personajes...) Este artículo divide el comportamiento de movimiento en tres niveles. Se centrará en el nivel medio de conductas de dirección, describirá brevemente el nivel inferior de locomoción y tocará ligeramente el nivel superior de establecimiento de objetivos y estrategia.

Introducción

Los personajes autónomos son un tipo de agente autónomo destinado a su uso en animación por computadora y medios interactivos como juegos y realidad virtual. Estos agentes representan un personaje de una historia o juego y tienen cierta capacidad para improvisar sus acciones. Esto contrasta tanto con un personaje de una película animada, cuyas acciones están escritas de antemano, como con un “avatar” en un juego o realidad virtual, cuyas acciones son dirigidas en tiempo real por un jugador o participante humano. En los juegos, los personajes autónomos a veces se denominan personajes no jugadores.

Un personaje autónomo debe combinar aspectos de un robot autónomo con algunas habilidades de un actor humano en teatro de improvisación. Estos personajes no suelen ser robots reales y ciertamente no son actores humanos, pero comparten algunas propiedades de cada uno.

El término “agente autónomo” se utiliza en muchos contextos, por lo que lo siguiente es un intento de ubicar la terminología de este artículo en relación con otros campos de estudio. Un agente autónomo puede existir de forma aislada o puede estar situado en un mundo compartido por otras entidades. Un agente de “minería de datos” es un ejemplo del primero, y un controlador de una red eléctrica es un ejemplo del segundo. Un agente situado puede ser reactivo (instintivo, impulsado por un estímulo) o puede ser deliberativo (“intelectual” en el sentido clásico de la IA). Un agente autónomo puede tratar exclusivamente con información abstracta (“softbot”, “knowbot” o “agente de información”) o puede encarnarse en una manifestación física (un típico robot industrial o un vehículo autónomo). Las combinaciones de situado, reactivo y encarnado definen varias clases distintas de agentes autónomos.

La categoría de agentes situados y encarnados suele sugerir robots autónomos: dispositivos mecánicos que existen en el mundo real. A veces los robots se estudian mediante simulación computacional. Pero los puristas en el campo de la robótica ven con recelo esa práctica porque la simulación puede divergir de la realidad de maneras impredecibles. Existe otra clase de agente situado y encarnado basado en un modelo computacional. Este artículo utilizará el término virtual.

(como en la realidad virtual) para denotar estos agentes que, en lugar de ser simulaciones de un dispositivo mecánico en el mundo real, son agentes reales en un mundo virtual. (Análogo a un modelo basado físicamente en animación por computadora.) Por lo tanto, los personajes autónomos del título de este artículo son: agentes virtuales situados, encarnados, reactivos.

El término comportamiento tiene muchos significados. Puede significar la acción compleja de un humano u otro animal basada en la voluntad o el instinto. Puede significar las acciones en gran medida predecibles de un sistema mecánico simple o la acción

compleja de un sistema caótico. En aplicaciones multimedia y de realidad virtual, a veces se utiliza como sinónimo de "animación". En este artículo se utiliza el término conducta para referirse a las acciones improvisadas y realistas de carácter autónomo.

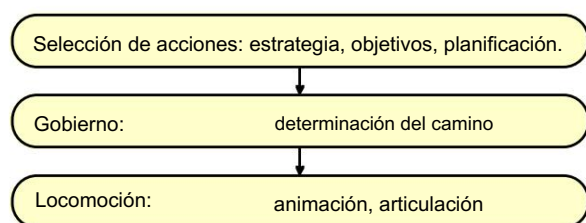


Figura 1: Una jerarquía de comportamientos de movimiento

El comportamiento de un personaje autónomo se puede entender mejor dividiéndolo en varias capas. Estas capas están destinadas únicamente a brindar claridad y especificidad en la discusión que seguirá. La Figura 1 muestra una división del comportamiento de movimiento de personajes autónomos en una jerarquía

de tres capas: selección de acción, dirección y locomoción. Ciertamente son posibles otras disecciones. Blumberg y Galyean [Blumberg 95] describen una jerarquía similar de tres capas, a las que llaman: motivación, tarea y motor. Tenga en cuenta que, si bien la jerarquía de comportamiento presentada aquí pretende ser ampliamente aplicable a comportamientos de movimiento, no es adecuada para otros tipos de acciones autónomas; por ejemplo, los comportamientos conversacionales de un "charlabot" requieren una estructura significativamente diferente.

Consideremos, por ejemplo, algunos vaqueros que cuidan un rebaño de ganado en el campo. Una vaca se aleja del rebaño. El jefe del sendero le dice a un vaquero que vaya a buscar al perro callejero. El vaquero le dice "maréate" a su caballo y lo guía hacia la vaca, posiblemente evitando obstáculos en el camino. En este ejemplo, el jefe del sendero representa la selección de acciones: notar que el estado del mundo ha cambiado (una vaca abandonó el rebaño) y establecer una meta (recuperar a la pérdida). la direccion

El nivel está representado por el vaquero, que descompone la meta en una serie de submetas simples (acercarse a la vaca, evitar obstáculos, recuperar la vaca). Un subobjetivo corresponde a un comportamiento de dirección del equipo de vaquero y caballo. Mediante diversas señales de control (órdenes vocales, espuelas, riendas), el vaquero dirige su caballo hacia el objetivo. En términos generales, estas señales expresan conceptos como: ir más rápido, ir más lento, girar a la derecha, girar a la izquierda, etcétera. El caballo implementa el nivel de locomoción. Tomando como entrada las señales de control del vaquero, el caballo se mueve en la dirección indicada. Este movimiento es el resultado de una compleja interacción entre la percepción visual del caballo, su sentido del equilibrio y sus músculos aplicando torsión a las articulaciones de su esqueleto. Desde el punto de vista de la ingeniería, la locomoción de las piernas es un problema muy difícil [Raibert 91], [Hodgins 95], pero ni el vaquero ni el caballo lo piensan dos veces.

Este artículo se centrará en la dirección, la capa intermedia de la jerarquía conductual. Describiré brevemente un modelo simple de la capa de locomoción, pero sólo con suficiente detalle para proporcionar una base concreta para la discusión de varios comportamientos de dirección. Habrá una breve discusión sobre la selección de acciones, pero principalmente en el contexto de combinar y mezclar comportamientos de dirección básicos.

La búsqueda de caminos es un tema relacionado con el tema de este artículo, pero separado de él. Los algoritmos de búsqueda de caminos como A* y Dijkstra operan en redes (que a menudo representan cuadrículas) y esencialmente resuelven laberintos. Esta solución podría servir como especificación de las técnicas de dirección descritas en este artículo. Una analogía podría ser comparar las instrucciones de conducción escritas para ir de un lugar a otro con el acto de conducir el coche por esa ruta.

Para obtener una excelente visión general de la búsqueda de caminos, consulte [Reese 99].

Para comprender el objetivo de este trabajo, cabe señalar que los comportamientos de dirección que se analizan aquí se relacionan con el movimiento "rápido": correr versus gatear. Esta es una noción informal, pero pretende sugerir que la velocidad típica de un personaje es grande en relación con su aceleración máxima de giro. Como resultado, los comportamientos directivos deben anticipar el futuro y tener en cuenta las posibles consecuencias de las acciones actuales.

Trabajo relacionado

Los comportamientos directivos de personajes autónomos se basan en una larga historia de investigaciones relacionadas en otros campos. Las máquinas autónomas, los servomecanismos y la teoría del control tienen sus raíces en la década de 1940, como se describe en el libro de Norbert Wiener de 1948 *Cibernética o control y comunicación en el animal y la máquina* [Wiener 48]. El término cibernética proviene de una palabra griega que significa timonel. A finales de los años 40, el neurofisiólogo Gray Walter construyó tortugas robóticas autónomas [Walter 50] que incorporaban varios de los comportamientos de dirección descritos aquí y estuvieron entre las primeras máquinas en exhibir un comportamiento emergente similar a la vida.

A principios de la década de 1980, Valentino Braitenberg extrapoló los prototipos de Walter a experimentos mentales sobre una serie de "vehículos" fantasiosos con comportamientos progresivamente más complejos [Braitenberg 84]. David Zeltzer comenzó a aplicar técnicas y modelos de inteligencia artificial a aplicaciones de animación [Zeltzer 83]. Y en 1987, creé un modelo de comportamiento animado de bandadas de aves utilizando técnicas estrechamente relacionadas con las presentadas en este artículo [Reynolds 87].

La siguiente lista de investigaciones relacionadas se divide en tres categorías generales: robótica, inteligencia artificial y vida artificial, aunque en algunos casos la distinción es algo arbitraria.

Generalmente estos trabajos están orientados hasta cierto punto hacia la animación: se ubican en la superposición entre la animación (o juegos, VR y multimedia) y estos otros tres campos.

Trabajos relacionados con la robótica. Rodney Brooks popularizó la noción entonces radical de construir controladores reactivos para sistemas robóticos [Brooks 85]. Aunque originalmente se inspiró en la investigación etológica (comportamiento animal), el trabajo de Ron Arkin [Arkin 87, 89, 92] se ha centrado en la aplicación de comportamientos de dirección a robots móviles. La investigación de Arkin ha sido paralela a gran parte del trabajo presentado en este artículo, pero sus esquemas (mapeos de percepción/acción) se expresan en términos de modelos de campo potenciales en contraposición al enfoque procedimental descrito aquí. En algunos casos se trata de una distinción sin diferencia, pero en otros casos (como la evitación de obstáculos) conduce a un comportamiento de agente significativamente diferente. Marc Raibert y Jessica Hodgins

Ambos comenzaron en la investigación de robótica con patas y ahora trabajan en aplicaciones de animación de sistemas de patas físicamente realistas. En ambos casos, su trabajo ha abordado aspectos de dirección y planificación de trayectoria de estos sistemas [Raibert 91, 91b], [Hodgins 95]. El trabajo de Zapata et al. sobre controladores de dirección para robots móviles rápidos se centró en estrategias que tenían que lidiar con el impulso y otros aspectos del movimiento mecánico rápido [Zapata 1992]. Maja Mataric ha trabajado extensamente en robótica colectiva [Mataric 93] y un tema central de este trabajo es la dirección.

Trabajos relacionados con la inteligencia artificial. Ken Kahn creó uno de los primeros sistemas que generaba animación del movimiento de los personajes a partir de descripciones de historias [Kahn 79]. David Zeltzer [Zeltzer 83, 90] fue pionero en la animación basada en IA, popularizando la idea de la especificación abstracta del movimiento a "nivel de tarea". Gary Ridsdale [Ridsdale 87] creó personajes capaces de improvisar movimientos complejos, yendo de A a B evitando obstáculos estáticos y otros actores. El trabajo Desktop Theatre de Steve Strassmann [Strassmann 1991] amplió estas nociones para incluir el manejo de accesorios y la representación emocional. El trabajo de animación conductual basado en agentes de Mônica Costa [Costa 90] permite a un personaje navegar por una casa evitando obstáculos de forma reactiva. Joseph Bates et al. están llevando a cabo una investigación sobre personajes dramáticos e improvisados, que aborda el comportamiento de dirección, en el Proyecto Oz (y ahora Zoesis) . [Bates 92] y en The Virtual Theatre Project de Barbara Hayes-Roth et al. [Hayes-Roth 96].

Trabajos relacionados con la vida artificial (y otros campos). El modelo boids de 1987 de rebaños, rebaños, escuelas y movimiento grupal relacionado [Reynolds 87], descompuso este complejo comportamiento grupal en tres comportamientos de dirección simples a nivel individual. Al año siguiente se presentaron comportamientos de dirección relacionados para evitar obstáculos [Reynolds 88]. En el Taller de Vida Artificial de 1987, Mitchel Resnick presentó trabajos sobre vehículos autónomos implementados en LEGO LOGO [Resnick 89] y Michael Travers demostró su kit de construcción animal AGAR [Travers 89]. (Ver también trabajos más recientes de estos autores [Resnick 93] y [Travers 94]).

Los comportamientos de dirección fueron un elemento clave en The Virtual Fishtank, una instalación de realidad virtual multiusuario en The Computer Museum creada por equipos del Media Lab del MIT y NearLife [Resnick 98]. Armin Bruderlin generó procedimentalmente una animación dirigida a objetivos del caminar humano [Bruderlin 1989]. La disertación de Randall Beer sobre una cucaracha artificial [Beer 90] es notable por la profundidad y complejidad de su modelo neuroetológico. Un elemento central de este modelo es la implementación neuronal de varios tropismos (como la quimiotaxis y la tigmotaxis) que son análogos directos de los comportamientos de dirección que se describen a continuación. En [Wilhelms 90] Jane Wilhelms y Robert Skinner investigan arquitecturas para personajes con apariencia de vehículos. Thalmann et al. Creó personajes de animación conductual que navegaban por pasillos y sorteaban obstáculos utilizando la visión simulada con renderizado 3D [Thalmann 90]. Michiel van de Panne creó controladores para tareas como el estacionamiento en paralelo de un automóvil mediante la búsqueda en el espacio de estados [van de Panne 90]. G. Keith Still ha modelado grandes multitudes humanas utilizando un modelo del comportamiento de dirección de cada individuo [Still 94]. Utilizando un algoritmo genético modificado, Karl Sims evolucionó simultáneamente cerebros y cuerpos para criaturas artificiales para diversos estilos de locomoción y para la búsqueda de objetivos [Sims 94]. En un trabajo reportado por primera vez en SAB94 y actualizado en SAB96 [Cliff 96], Cliff y Miller coevolucionaron comportamientos de persecución y evasión para agentes depredadores y presas. Xiaoyuan Tu et al. desarrollaron un modelo elaborado y sorprendentemente realista de la biomecánica, locomoción, percepción y comportamiento de los peces en [Tu 94, 96] que incluía locomoción física, comportamientos de dirección y un sistema etológico para la selección de acciones. En [Blumberg 94] Bruce Blumberg describió un mecanismo detallado para complejos

selección de acción y con Tinsley Galyean en [Blumberg 95] discutieron el diseño de un personaje de realidad virtual capaz tanto de improvisación autónoma como de respuesta a la dirección externa. Una aplicación de estos caracteres fue en el sistema ALIVE [Maes 95] de Patties Maes et al. El sistema de improvisación de Ken Perlin y Athomas Goldberg [Perlin 96] también cubre la gama desde la locomoción hasta la selección de acciones, pero utiliza un enfoque único basado en secuencias de comandos conductuales y la síntesis procesal de texturas de Perlin de 1985 [Perlin 85] aplicada al movimiento. James Cremer y sus colegas han creado conductores autónomos para que sirvan como "extras" creando tráfico ambiental en simuladores interactivos de conducción de automóviles [Cremer 96]. Robin Green (de Bullfrog/EA) ha desarrollado un sistema maduro para personajes autónomos utilizado en Dungeon Keeper 2 que se inspiró en parte en un borrador inicial de este artículo. Dave Pottinger ha proporcionado una discusión detallada sobre la dirección y coordinación de grupos de personajes en juegos [Pottinger 1999].

Locomoción

La locomoción es la parte inferior de la jerarquía conductual de tres niveles descrita anteriormente. La capa de locomoción representa la encarnación de un personaje. Convierte las señales de control de la capa de dirección en movimiento del "cuerpo" del personaje. Este movimiento está sujeto a restricciones impuestas por el modelo físico del cuerpo, como la interacción del impulso y la fuerza (limitación de las fuerzas que puede aplicar el cuerpo).

Como se describió anteriormente, el caballo de un vaquero puede considerarse como un ejemplo de capa de locomoción. Las decisiones de dirección del jinete se transmiten mediante simples señales de control al caballo, quien las convierte en movimiento. El objetivo de hacer una distinción abstracta entre dirección y locomoción es anticipar "conectar" un nuevo módulo de locomoción. Imagínese levantar al jinete del caballo y colocarlo en una motocicleta de cross-country. La selección de objetivos y el comportamiento de dirección siguen siendo los mismos. Lo único que ha cambiado es el mecanismo para poner en movimiento las señales de control (ir más rápido, girar a la derecha,...). Originalmente implicaba locomoción con piernas (equilibrio, huesos, músculos) y ahora implica locomoción con ruedas (motor, ruedas, frenos). El papel del jinete no cambia.

Esto sugiere que con una convención adecuada para comunicar señales de control, los comportamientos de dirección pueden ser completamente independientes del esquema de locomoción específico. Aunque en la práctica es necesario compensar la "agilidad" y las diferentes "características de manejo" de los sistemas de locomoción individuales. Esto se puede hacer ajustando los parámetros de ajuste para un esquema de locomoción determinado (que es el enfoque adoptado en los comportamientos de dirección que se describen a continuación) o utilizando una técnica adaptativa y de autocalibración (la forma en que un conductor humano se adapta rápidamente a las características de un vehículo desconocido). automóvil). En el primer caso, un comportamiento de dirección podría determinar a través de su ajuste a priori que la velocidad del personaje en una situación dada debería ser de 23 mph, en el segundo caso podría decir "reduzca la velocidad un poco" hasta obtener el mismo resultado.

La locomoción de un personaje autónomo puede basarse en su representación animada o ser independiente de ella. Un personaje podría representarse mediante una simulación de caminar dinámicamente equilibrada basada en lo físico, proporcionando tanto animación realista como locomoción conductual.

O un personaje puede tener un modelo de locomoción muy simple (como el que se describe en la siguiente sección) al que se adjunta una representación estática (por ejemplo, una nave espacial) o preanimada (como una figura humana realizando un ciclo de caminata). Un enfoque híbrido consiste en utilizar un modelo de locomoción simple y un modelo de animación adaptativo, como un ciclo de caminata impulsado por cinemática inversa, para cerrar la brecha entre la locomoción abstracta y el terreno concreto. Finalmente, la locomoción puede restringirse al movimiento.

inherente a un conjunto fijo de segmentos preanimados (caminar, correr, detenerse, girar a la izquierda...) que se seleccionan discretamente o se combinan.

Un modelo de vehículo simple

El enfoque adoptado en este artículo es considerar los comportamientos de dirección como esencialmente independientes del esquema de locomoción subyacente. Se presentará un modelo de locomoción simple para hacer más concreta la discusión sobre los comportamientos de dirección. Este modelo de locomoción se basará en un vehículo simple idealizado. La elección del término "vehículo" se inspira en cierta medida en [Braitenberg 84]. Su objetivo es abarcar una amplia gama de medios de transporte, desde dispositivos con ruedas hasta caballos, desde aviones hasta submarinos y (aunque ciertamente amplía la terminología) incluir la locomoción mediante las propias piernas de un personaje. El modelo de vehículo descrito aquí es tan simplista y genérico que es una aproximación igualmente buena (o igualmente mala) a todos ellos.

Este modelo de vehículo se basa en una aproximación de masa puntual. Por un lado, eso permite un modelo basado en física muy simple y computacionalmente barato (por ejemplo, una masa puntual tiene velocidad (momento lineal) pero no momento de inercia (momento de rotación)). Por otro lado, no puede ser un modelo físico muy convincente porque las masas puntuales no existen en el mundo real. Cualquier objeto físico con masa debe tener un radio distinto de cero y, por tanto, un momento de inercia. Este uso de un modelo de vehículo no físico excesivamente simplificado es meramente por conveniencia y pretende ser "sin pérdida de generalidad"; siempre debería ser posible sustituirlo por un modelo de vehículo más plausible y realista con base física.

Una masa puntual está definida por una propiedad de posición y una propiedad de masa. Además, el modelo de vehículo simple incluye una propiedad de velocidad. La velocidad se modifica aplicando fuerzas.

Como se trata de un vehículo, estas fuerzas generalmente son autoaplicadas y, por tanto, limitadas. Por ejemplo, una fuerza típica que ajusta la velocidad de un vehículo es el empuje, generado por la propia central eléctrica del vehículo y, por tanto, limitado en magnitud por la capacidad de la central eléctrica. Para el modelo de vehículo simple, esta noción se resume en un único parámetro de "fuerza máxima" (`max_force`). La mayoría de los vehículos se caracterizan por una velocidad máxima. Normalmente, esta limitación se debe a la interacción entre la aceleración debida a su empuje finito y la desaceleración debida al arrastre viscoso, la fricción o (en sistemas con patas) el impulso de las piezas alternativas. Como alternativa a la simulación realista de todas estas fuerzas limitantes, el modelo de vehículo simple incluye un parámetro de "velocidad máxima" (`max_speed`). Este límite de velocidad se impone mediante un truncamiento cinemático del vector de velocidad del vehículo. Finalmente, el modelo de vehículo simple incluye una orientación que, junto con la posición del vehículo, forma un espacio de coordenadas local alineado con la velocidad al que se puede unir un modelo geométrico del vehículo. (Los términos localizar y globalizar se utilizarán en este artículo para connotar vectores de transformación dentro y fuera de este espacio local).

Modelo de vehículo sencillo:

- masa
- escalar
- vector de posición
- vector de velocidad
- escalar `max_force`
- escalar `max_speed`
- vectores de base N de orientación

Para un modelo de vehículo 3D, los valores de los vectores de posición y velocidad tienen tres componentes y el valor de orientación es un conjunto de tres vectores (o una matriz de 3x3 o un cuaternión). Para un vehículo 2D, cada uno de los vectores tiene dos componentes y el valor de orientación son dos vectores base 2D o se puede representar como un único ángulo de rumbo escalar.

La física del modelo de vehículo simple se basa en la integración directa de Euler. En cada paso de la simulación, se aplican fuerzas de dirección determinadas por el comportamiento (limitadas por `max_force`) a la masa puntual del vehículo. Esto produce una aceleración igual a la fuerza de dirección dividida por la masa del vehículo. Esa aceleración se suma a la velocidad anterior para producir una nueva velocidad, que luego se trunca mediante `max_speed`. Finalmente, la velocidad se suma a la posición anterior:

```
fuerza_dirección = truncar (dirección_dirección, fuerza_máxima)
aceleración = fuerza_dirección / masa
velocidad = truncar (velocidad + aceleración, max_speed)
posición = posición + velocidad
```

El modelo de vehículo simple mantiene su espacio local alineado con la velocidad mediante un ajuste incremental. desde el paso de tiempo anterior. El sistema de coordenadas local se define en términos de cuatro vectores: un vector de posición que especifica el origen local y tres vectores de dirección que sirven como vectores base del espacio. Los vectores base indican la dirección y la longitud de las unidades de coordenadas en cada una de tres direcciones mutuamente perpendiculares con respecto al vehículo. A estos ejes nos referiremos aquí como adelante, arriba y lateral. (Estos corresponden, por supuesto, a los ejes X, Y y Z de R

³. Pero algunas personas piensan que obviamente es Y, mientras que otras piensan que es obviamente Z. El método descriptivo Se utilizarán términos en lugar de los nombres cartesianos para mayor claridad.)

Para permanecer alineados con la velocidad en cada paso de tiempo, los vectores base deben rotarse en una nueva dirección. (Si la velocidad es cero, se conserva la antigua orientación). En lugar de utilizar rotaciones explícitas, el espacio local se reconstruye utilizando una combinación de sustitución, aproximación y reortogonalización. Comenzamos con la nueva velocidad y una aproximación a la nueva velocidad.

dirección. Por ejemplo, la antigua dirección ascendente se puede utilizar como una aproximación a la nueva dirección ascendente. Usamos la operación de producto vectorial vectorial para construir los nuevos vectores base:

```
new_forward = normalizar (velocidad)
aproximado_arriba = normalizar (aproximado_arriba) // si es necesario
nuevo_lado = cruzar (nuevo_adelante, aproximado_arriba)
nuevo_arriba = cruz (nuevo_adelante, nuevo_lado)
```

La idea básica es que la subida aproximada es casi perpendicular a la nueva dirección de avance, porque los cambios de orientación de cuadro a cuadro suelen ser pequeños. La nueva dirección lateral será perpendicular al nuevo avance, según la definición de producto cruzado. El nuevo up es el producto cruzado de la perpendicular hacia adelante y hacia el lado y por lo tanto es perpendicular a cada uno.

El concepto de "alineación de velocidades" no especifica únicamente una orientación. El grado de libertad correspondiente a la rotación alrededor del eje delantero (también conocido como balanceo) permanece sin restricciones. La construcción del nuevo espacio local con respecto al anterior (utilizando, por ejemplo, la antigua dirección ascendente como aproximación inicial a la nueva) garantizará que la orientación del rollo al menos permanezca constante. Definir el valor de balanceo "correcto" requiere heurísticas adicionales, basadas en el uso previsto del modelo de vehículo.

Para un vehículo "volador" (como aviones, naves espaciales y submarinos) es útil definir el balanceo en términos de inclinación. La idea básica de la inclinación es alinear el "piso" del vehículo (eje arriba) con la gravedad aparente debida a la fuerza centrífuga durante un giro. Por el contrario, queremos que la dirección ascendente sea

alinearse con la fuerza centrípeta que produjo la maniobra. En presencia de gravedad, la dirección hacia abajo debe alinearse con la suma de la aceleración de giro y la aceleración gravitacional.

También queremos agregar la orientación actual para amortiguar cambios abruptos en el balanceo. Entonces, para implementar la inclinación en el modelo de vehículo simple, la dirección ascendente aproximada es una suma ponderada de: aceleración de la dirección, aceleración gravitacional y la anterior .

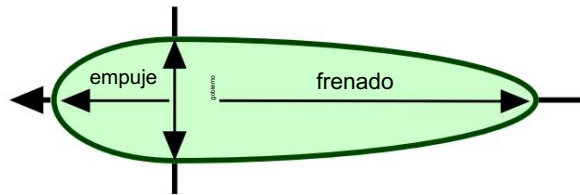


Figura 2: fuerzas de dirección asimétricas

Para un vehículo que "se adapta a la superficie" (con ruedas, deslizante o con patas), queremos restringir la posición del vehículo con respecto a la superficie y alinear el eje superior del vehículo con la normal a la superficie. Además, la velocidad debe limitarse a ser puramente tangencial a la superficie. Estos requisitos pueden cumplirse fácilmente si el colector de superficie se representa de tal manera que un punto arbitrario en el espacio (correspondiente a la antigua posición del vehículo) pueda asignarse a: (1) el punto más cercano en la superficie, y (2) el superficie normal en ese punto. La velocidad se puede hacer tangente restando la porción normal a la superficie. La posición del vehículo se establece en el punto de la superficie y la normal a la superficie se convierte en su eje superior .

En este sencillo modelo de vehículo, la señal de control transmitida desde las conductas de dirección a las conductas de locomoción consta exactamente de una cantidad vectorial: una fuerza de dirección deseada. Los modelos de vehículos más realistas tendrían conjuntos de señales de control muy diferentes. Por ejemplo, un automóvil tiene un volante, un acelerador y un freno, cada uno de los cuales puede representarse como cantidades escalares. Es posible mapear un vector de fuerza de dirección generalizado en estas señales escalares: el componente lateral del vector de dirección puede interpretarse como la señal de dirección, el componente delantero del vector de dirección puede mapearse en la señal del acelerador si es positiva, o en la señal de freno si es negativa. Estas asignaciones pueden ser asimétricas; por ejemplo, un automóvil típico puede desacelerar debido al frenado mucho más rápido de lo que puede acelerar debido al empuje del motor, como se muestra en la Figura 2.

Debido a su supuesto de alineación de velocidad, este modelo de vehículo simple no puede simular efectos como derrapes, giros o deslizamientos. Además este modelo permite que el vehículo gire cuando su velocidad es cero. La mayoría de los vehículos reales no pueden hacer esto (son "no holonómicos") y en cualquier caso

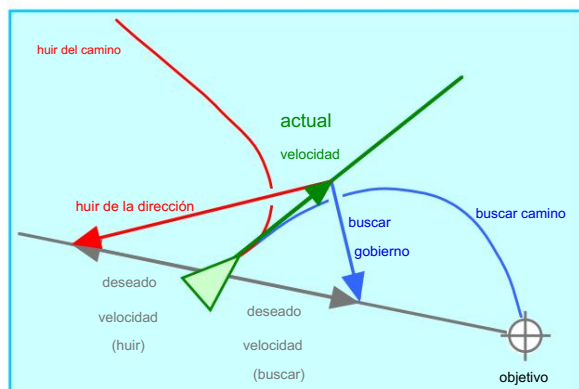


Figura 3: buscar y huir

permite cambios de orientación indeseablemente grandes durante un único paso de tiempo. Este problema se puede resolver imponiendo una restricción adicional al cambio de orientación, o limitando el componente de dirección lateral a bajas velocidades, o simulando un momento de inercia.

Comportamientos de dirección

Esta discusión sobre comportamientos de dirección específicos supone que la locomoción se implementa mediante el modelo de vehículo simple descrito anteriormente y está parametrizada por un único vector de fuerza de dirección. Por lo tanto, los comportamientos de dirección se describen en

permite cambios de orientación indeseablemente grandes durante un único paso de tiempo. Este problema se puede resolver imponiendo una restricción adicional al cambio de orientación, o limitando el componente de dirección lateral a bajas velocidades, o simulando un momento de inercia.

Comportamientos de dirección

Comportamientos de dirección

Esta discusión sobre comportamientos de dirección específicos supone que la locomoción se implementa mediante el modelo de vehículo simple descrito anteriormente y está parametrizada por un único vector de fuerza de dirección. Por lo tanto, los comportamientos de dirección se describen en

términos del cálculo geométrico de un vector que representa una fuerza de dirección deseada. Tenga en cuenta que, en general, la magnitud de estos vectores de dirección es irrelevante, ya que normalmente el modelo del vehículo los recortará a `max_force`. Tenga en cuenta también que muchas de las llamadas a funciones de longitud y normalización en estas formulaciones pueden reemplazarse por rutinas rápidas que utilizan una aproximación a la longitud como en [Ohashi 94]. Los términos "nosotros" o "nuestro" a veces se utilizarán para indicar la perspectiva en primera persona del personaje que está siendo guiado por un comportamiento determinado. Se pueden encontrar diagramas animados que ilustran estos comportamientos en la web en <http://www.red.com/cwr/steer/>

La búsqueda (o persecución de un objetivo estático) actúa para dirigir al personaje hacia una posición específica en el espacio global. Este comportamiento ajusta el personaje para que su velocidad esté alineada radialmente hacia el objetivo. Tenga en cuenta que esto es diferente de una fuerza de atracción (como la gravedad) que produciría una trayectoria orbital alrededor del punto objetivo. La "velocidad deseada" es un vector en la dirección del personaje al objetivo. La longitud de la "velocidad deseada" podría ser `max_speed`, o podría ser la velocidad actual del personaje, dependiendo de la aplicación particular. El vector de dirección es la diferencia entre esta velocidad deseada y la velocidad actual del personaje, consulte la Figura 3.

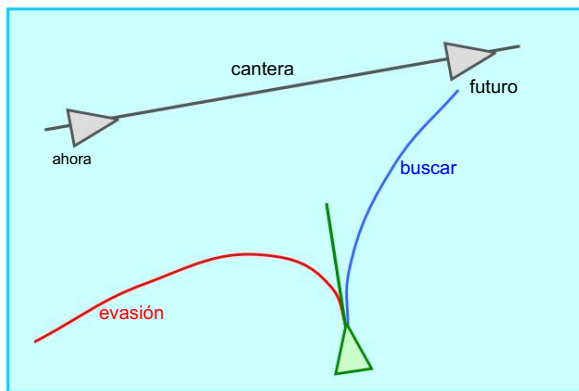


Figura 4: persecución y evasión

$$\text{velocidad_deseado} = \text{normalizar}(\text{posición} - \text{objetivo}) * \text{velocidad_máxima}$$
$$\text{dirección} = \text{velocidad_deseado} - \text{velocidad}$$

Si un personaje continúa buscando, eventualmente atravesará el objetivo y luego regresará para acercarse nuevamente. Esto produce un movimiento similar al de una polilla zumbando alrededor de una bombilla. Compare esto con la descripción de la llegada a continuación.

Huir es simplemente lo inverso de buscar y actúa para dirigir al personaje de modo que su velocidad esté alineada radialmente lejos del objetivo. La velocidad deseada apunta en la dirección opuesta.

La persecución es similar a la búsqueda excepto que la presa (objetivo) es otro personaje en movimiento. La persecución eficaz requiere una predicción de la posición futura del objetivo. El enfoque adoptado aquí es utilizar un predictor simple y reevaluarlo en cada paso de la simulación. Por ejemplo, un predictor lineal basado en la velocidad corresponde al supuesto de que la cantera no girará durante el intervalo de predicción. Si bien esta suposición suele ser incorrecta, la predicción resultante solo se utilizará durante aproximadamente 1/30 de segundo. La posición de un personaje T unidades de tiempo en el futuro (suponiendo que no maniobre) se puede obtener escalando su velocidad en T y sumando ese desplazamiento a su posición actual. La dirección para la persecución es entonces simplemente el resultado de aplicar el comportamiento de dirección de búsqueda a la ubicación objetivo prevista.

Ver Figura 4.

La clave para esta implementación de la persecución es el método utilizado para estimar el intervalo de predicción T . Idealmente, T sería el tiempo hasta la intercepción, pero ese valor es imposible de conocer porque la cantera puede realizar maniobras arbitrarias e impredecibles. Se podría suponer que T es una constante, lo que, si bien es ingenuo, produciría una mejor búsqueda que una simple búsqueda (que corresponde a $T=0$). Sin embargo, para un rendimiento razonable, T debería ser mayor cuando el perseguidor está lejos de la presa y pequeño cuando está cerca. Un estimador simple de

la calidad moderada es $T=Dc$ donde D es la distancia entre el perseguidor y la presa, y c es un parámetro de giro. Se puede obtener un estimador más sofisticado teniendo en cuenta los rumbos relativos del perseguidor y la cantera, y si el perseguidor está generalmente delante, detrás o al lado de la cantera. Estas dos métricas se pueden expresar en términos de productos escalares simples (entre vectores unitarios de avance y entre el avance de la presa y el desplazamiento a la posición del perseguidor).

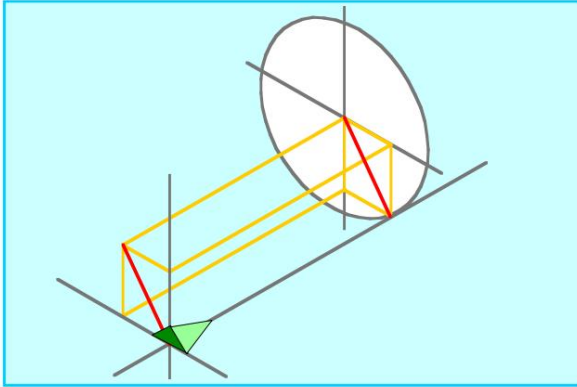


Figura 5: persecución compensada

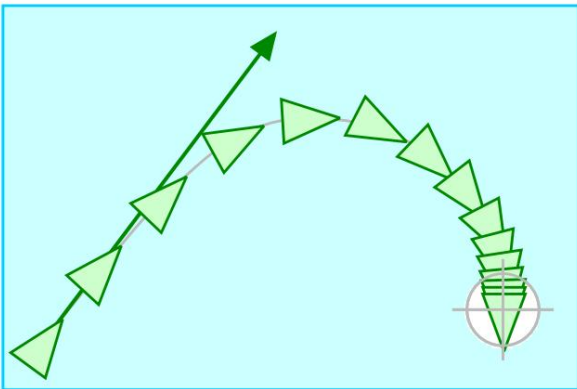


Figura 6: Llegada

Tenga en cuenta que se debe tener cuidado de reducir T (por ejemplo, a cero) cuando el perseguidor se encuentra alineado con su presa y delante de ella.

Otro enfoque tanto para la búsqueda como para la persecución se basa en el hecho de que cuando nuestro personaje está en curso de colisión con un objetivo, este aparecerá en un rumbo constante en el espacio local de nuestro personaje.

Por el contrario, nuestro personaje puede dirigirse hacia la intercepción logrando mantener al objetivo en un rumbo constante.

La evasión es análoga a la persecución, excepto que huir se utiliza para alejarse de la posición futura prevista del personaje objetivo. Existen técnicas óptimas para la persecución y la evasión en el campo de la teoría del control [Isaacs 65]. Las versiones proporcionadas aquí pretenden ser ligeras y no óptimas. En los sistemas naturales, la evasión es a menudo "intencionalmente" no óptima para ser impredecible, lo que le permite frustrar las estrategias de búsqueda predictivas, ver [Cliff 96].

La persecución compensada se refiere a dirigir un camino que pasa cerca, pero no directamente, de un objetivo en movimiento.

Algunos ejemplos serían una nave espacial que realiza un "sobrevuelo" o un avión que realiza una "carrera de ametrallamiento": volar lo suficientemente cerca como para estar dentro del alcance del sensor o del arma.

sin chocar con el objetivo. La idea básica es calcular dinámicamente un punto objetivo que esté desplazado por un radio dado R desde la posición futura prevista de la cantera, y luego usar el comportamiento de búsqueda para acercarse a ese punto desplazado, consulte la Figura 5. Para construir el punto desplazado: localizar la ubicación del objetivo prevista (en el espacio de coordenadas local de nuestro personaje) proyecta el objetivo local en el plano lateral del personaje, normaliza ese desplazamiento lateral, escale en $-R$, agrégalo al punto del objetivo local y globaliza ese valor.

El comportamiento de llegada es idéntico al de búsqueda mientras el personaje está lejos de su objetivo. Pero en lugar de moverse a través del objetivo a toda velocidad, este comportamiento hace que el personaje disminuya la velocidad a medida que se acerca al objetivo, eventualmente desacelerando hasta detenerse coincidente con el objetivo, como se muestra en la Figura 6. La distancia a la que comienza la desaceleración es un parámetro del comportamiento. Esta implementación es similar a buscar: se determina una velocidad deseada apuntando desde el personaje hacia el objetivo. Fuera del radio de parada, esta velocidad deseada se recorta a max_speed , dentro del radio de parada, la velocidad deseada se reduce (por ejemplo, linealmente) a cero.

```

target_offset = objetivo - posición
distancia = longitud (desplazamiento_objetivo)
velocidad_rampa = velocidad_máxima * (distancia / distancia_deceleración)
velocidad_recortada = mínima (velocidad_rampada, velocidad_máxima)
velocidad_deseado = (velocidad_recortada / distancia) * desplazamiento_objetivo
dirección = velocidad_deseado - velocidad

```

Ejemplos de este comportamiento en el mundo real incluyen a un jugador de béisbol que corre y luego se detiene en una base; o un automóvil que se dirige a una intersección y se detiene en un semáforo.

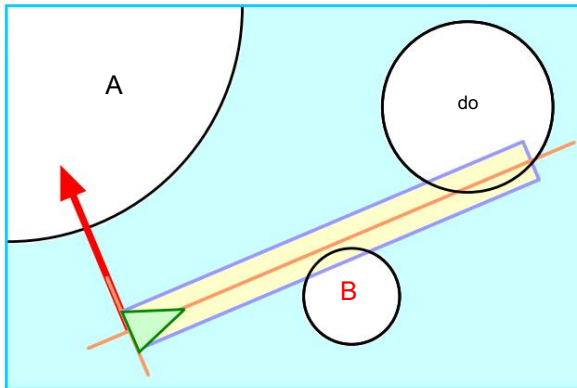


Figura 7: evitación de obstáculos

El comportamiento para evitar obstáculos le da al personaje la capacidad de maniobrar en un entorno desordenado esquivando obstáculos. Existe una distinción importante entre evitar obstáculos y huir. Huir siempre hará que un personaje se aleje de una ubicación determinada, mientras que evitar obstáculos solo se activa cuando un obstáculo cercano se encuentra directamente frente al personaje. Por ejemplo, si un automóvil circulaba paralelo a una pared, evitar obstáculos

no tomaría ninguna medida correctiva, sino que huiría. Intentaría alejarse de la pared y eventualmente conduciría perpendicular a ella.

La implementación del comportamiento para evitar obstáculos que se describe aquí hará una suposición simplificada de que tanto el carácter como el obstáculo pueden aproximarse razonablemente como esferas, aunque el concepto básico puede extenderse fácilmente a modelos de forma más precisos. Tenga en cuenta que esto se relaciona con evitar obstáculos, no necesariamente con la detección de colisiones. Imagínesse un avión intentando evitar una montaña. Ninguno de los dos tiene forma esférica, pero bastaría con que la esfera delimitadora del avión evite la esfera delimitadora de la montaña. Se puede utilizar una jerarquía descomponible de esferas delimitadoras para una representación eficiente de formas para la detección de colisiones [Hubbard 96] y, presumiblemente, también para evitar obstáculos. En [Egbert 96] se describe una técnica para evitar obstáculos no relacionada.

La construcción geométrica del comportamiento de evitación de obstáculos tiene cierta similitud con el comportamiento de persecución compensada descrito anteriormente. Es conveniente considerar la situación geométrica desde el sistema de coordenadas local del personaje. El objetivo del comportamiento es mantener un cilindro imaginario de espacio libre frente al personaje. El cilindro se encuentra a lo largo de la parte delantera del personaje. eje, tiene un diámetro igual a la esfera delimitadora del personaje y se extiende desde el centro del personaje por una distancia basada en la velocidad y agilidad del personaje. Un obstáculo más lejos que esta distancia no es una amenaza inmediata. El comportamiento para evitar obstáculos considera cada obstáculo por turno (quizás usando un esquema de división espacial para eliminar los obstáculos distantes) y determina si se cruzan con el cilindro. Al localizar el centro de cada obstáculo esférico, la prueba de no intersección con el cilindro es muy rápida. El centro del obstáculo local se proyecta en el plano lateral hacia arriba (estableciendo su coordenada delantera en cero) si la distancia 2D desde ese punto hasta el origen local es mayor que la suma de los radios del obstáculo y el personaje, entonces hay sin posible colisión. De manera similar, los obstáculos que están completamente detrás del personaje o completamente delante del cilindro se pueden rechazar rápidamente. Para cualquier

obstáculos restantes se realiza un cálculo de intersección línea-esfera. El obstáculo que cruza el eje delantero más cercano al personaje se selecciona como el "más amenazante".

La dirección para evitar este obstáculo se calcula negando la proyección lateral (lateral) hacia arriba del centro del

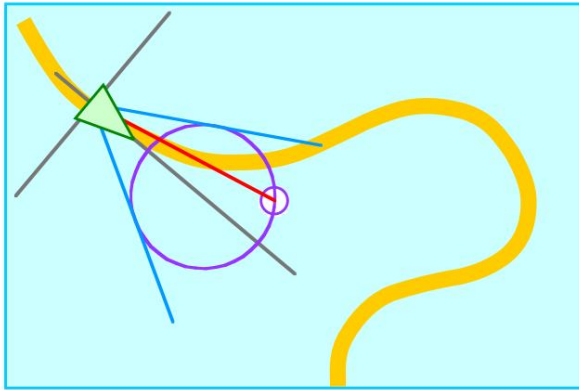


Figura 8: deambular

obstáculo. En la Figura 7, el obstáculo A no cruza el cilindro,

los obstáculos B y C sí, B se selecciona para evitarlo y la dirección correctiva está a la izquierda del personaje. El

valor devuelto al evitar obstáculos es (a) el valor de dirección para evitar el obstáculo más amenazante, o (b) si no hay colisión inminente, un valor especial (un valor nulo o el vector cero) para indicar que no hay dirección correctiva. se requiere en este momento.

Una nota final sobre la interacción entre evitar obstáculos

y buscar metas. Generalmente sólo nos preocupamos por los obstáculos que se encuentran entre nosotros y nuestro objetivo.

El avión ignora la montaña más allá del aeropuerto, pero la montaña

entre el avión y el aeropuerto es muy importante.

Wander es un tipo de dirección aleatoria. Una implementación sencilla sería generar una fuerza de dirección aleatoria en cada cuadro, pero esto produce un movimiento poco interesante. Es "nervioso" y no produce giros sostenidos. Un enfoque más interesante es mantener el estado de dirección de dirección y realizar pequeños desplazamientos aleatorios en cada cuadro. Por lo tanto, en un cuadro el personaje puede estar girando hacia arriba y hacia la derecha, y en el siguiente cuadro seguirá girando casi en la misma dirección. La fuerza de dirección realiza un "camino aleatorio" de una dirección a otra. Esta idea se puede implementar de varias maneras, pero una que ha producido buenos resultados es limitar la fuerza de dirección a la superficie de una esfera ubicada ligeramente por delante del personaje. Para producir la fuerza de dirección para el siguiente cuadro: se agrega un desplazamiento aleatorio al valor anterior y la suma se

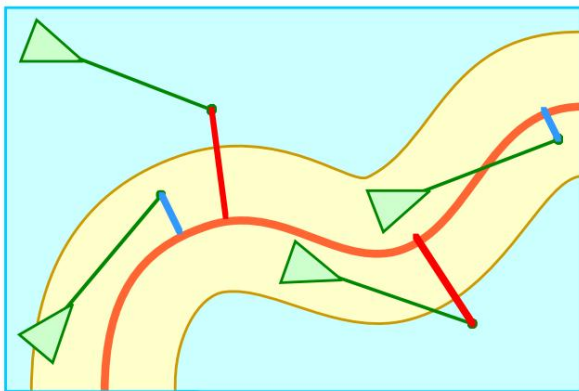


Figura 9: seguimiento del camino

limita nuevamente a la superficie de la esfera. El radio de la esfera (el círculo grande en la Figura 8) determina la "fuerza" máxima de desviación y la magnitud del desplazamiento aleatorio (el círculo pequeño en la Figura 8) determina la "tasa" de desviación.

Otra forma de implementar la desviación sería utilizar ruido

Perlin coherente [Perlin 85] para generar la dirección de dirección.

Relacionado con deambular está explorar (donde el objetivo es cubrir exhaustivamente una región del espacio) y buscar comida. (combinando deambulación con búsqueda de recursos).

Consulte [Beer 90] y [Tu 96] para obtener más detalles.

El comportamiento de seguimiento de camino permite a un personaje conducir a lo largo de un camino predeterminado, como una carretera, un corredor o un túnel. Esto es distinto de limitar rigidamente un vehículo a un camino como un

tren rodando por una vía. Más bien, el comportamiento de seguir un camino tiene como objetivo producir un movimiento como el de las personas que se mueven por un corredor: los caminos individuales permanecen cerca, y a menudo paralelos a, la línea central del corredor, pero son libres de desviarse de ella. En la implementación descrita aquí, una ruta se idealizará como una columna y un radio. La columna vertebral podría representarse como una curva spline o una "polilínea" (una serie de segmentos de línea conectados). El camino es entonces un "tubo" o "cilindro generalizado": un círculo del radio especificado, barrido a lo largo de la columna especificada. El objetivo del comportamiento de dirección que sigue el camino es mover a un personaje a lo largo del camino mientras permanece dentro del radio especificado de la columna. Si el personaje está inicialmente lejos del camino, primero debe acercarse y luego seguir el camino.

Para calcular la dirección para seguir la ruta, se hace una predicción basada en la velocidad de la dirección del personaje.

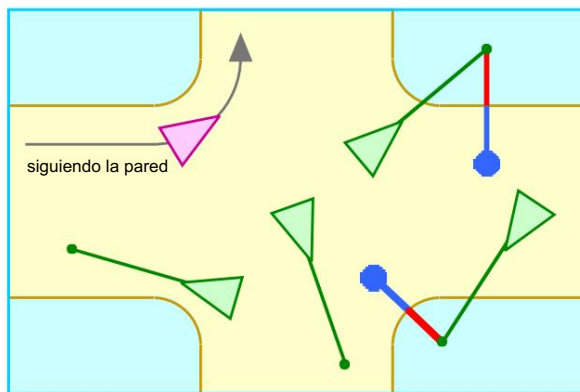


Figura 10: seguimiento de muro, contención

posición futura, como se discutió anteriormente con respecto al comportamiento de evitación de obstáculos. La posición futura prevista se proyecta en el punto más cercano de la columna vertebral del camino. Consulte la Figura 9. Si esta distancia de proyección (desde la posición prevista hasta el punto más cercano en el camino) es menor que el radio del camino, entonces se considera que el personaje está siguiendo correctamente el camino y no se requiere dirección correctiva. De lo contrario, el personaje se desvía del camino o está demasiado lejos del camino. Para volver a la trayectoria, se utiliza el comportamiento de búsqueda para dirigirse hacia la proyección en la trayectoria de la posición futura prevista. Al igual que en la evitación de obstáculos, se devuelve un valor nulo o cero si no se requiere dirección correctiva. Se puede seguir una ruta sin importar la dirección, o en una dirección específica (de A a B o de B a A) ajustando el punto objetivo a lo largo de la ruta en la dirección deseada.

Las variaciones en el seguimiento del camino incluyen el seguimiento de paredes y la contención, como se muestra en la Figura 10. Seguir una pared significa acercarse a una "pared" (u otra superficie o camino) y luego mantener un cierto desplazamiento respecto de ella [Beer 90]. Para una discusión sobre los objetivos de compensación, consulte búsqueda de compensación más arriba.

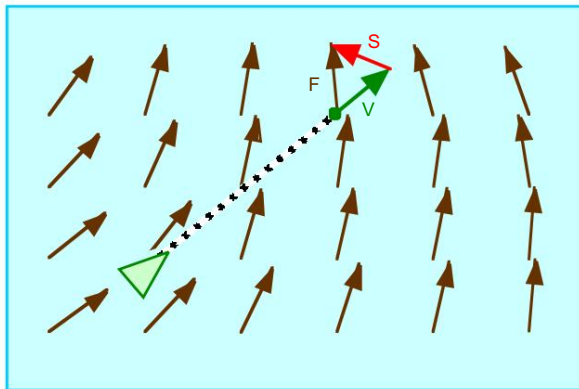


Figura 11: seguimiento del flujo

La contención se refiere al movimiento que está restringido a permanecer dentro de una determinada región. El seguimiento del camino es un tipo de contención donde la región permitida es un cilindro alrededor de la columna vertebral del camino. Ejemplos de contención incluyen: peces nadando en un acuario y jugadores de hockey patinando dentro de una pista de hielo. Para implementar: primero predice la posición futura de nuestro personaje, si está dentro de la región permitida no es necesaria ninguna dirección correctiva. De lo contrario nos dirigimos hacia la región permitida. Esto se puede lograr usando la búsqueda con un punto interno (por ejemplo, podemos proyectar la posición futura a la superficie del obstáculo y luego extender este desplazamiento para obtener un

punto objetivo) o podemos determinar la intersección de nuestro camino con el límite, encontrar la superficie normal en ese punto y luego usar el componente de la superficie normal que es perpendicular a nuestra dirección de avance como dirección lateral correctiva.

El campo de flujo que sigue el comportamiento de dirección proporciona una herramienta útil para dirigir el movimiento de los personajes en función de su posición dentro de un entorno. Es particularmente valioso en algunos equipos de producción porque permite realizar especificaciones de movimiento sin el uso de

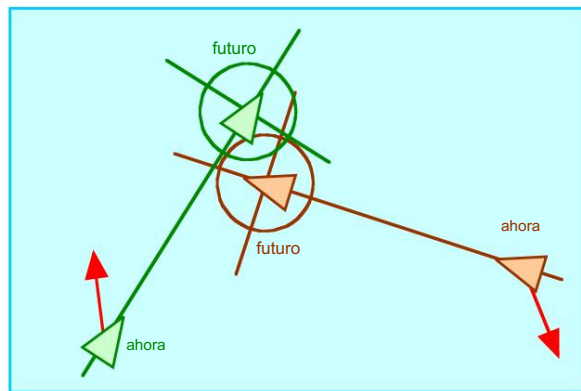


Figura 12:
evitación de colisiones no alineadas

programación y por lo tanto puede ser utilizado por el personal de arte directamente. En el caso de la producción de juegos, esta persona podría ser un "diseñador de niveles" y en la producción de animación podría ser un "planificador de escenas" o un "maquetador".

En el comportamiento de seguimiento del campo de flujo, el personaje gira para alinear su movimiento con la tangente local de un campo de flujo (también conocido como campo de fuerza o campo vectorial). El campo de flujo define un mapeo desde una ubicación en el espacio a un vector de flujo: imagine, por ejemplo, un piso con flechas pintadas. Un artista puede crear fácilmente un mapa de este tipo, que normalmente representa el plano de un entorno, con un programa de "pintura" especial que le permite dibujar el flujo de tráfico deseado con un pincel. La implementación del campo de flujo.

lo siguiente es muy simple. Se estima la posición futura de un carácter y se muestrea el campo de flujo en esa ubicación. Esta dirección del flujo (vector F en la Figura 11) es la "velocidad deseada" y la dirección de dirección (vector S) es simplemente la diferencia entre la velocidad actual (vector V) y la velocidad deseada.

El comportamiento de evitación de colisiones no alineado intenta evitar que los personajes que se mueven en direcciones arbitrarias choquen entre sí. Considere su propia experiencia al caminar por una plaza o vestíbulo lleno de otras personas que caminan: evitar colisiones implica predecir colisiones potenciales y alterar su dirección y velocidad para evitarlas. Si todos los personajes cercanos están alineados, se puede utilizar una estrategia menos complicada; consulte la separación a continuación.

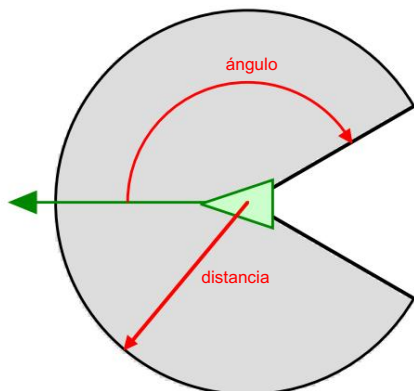


Figura 13: barrio

Para implementar esto como un comportamiento de dirección, nuestro personaje considera a cada uno de los demás personajes y determina (basándose en las velocidades actuales) cuándo y dónde los dos se acercarán más.

Existe una posibilidad de colisión si la aproximación más cercana se produce en el futuro y si la distancia entre los caracteres en la aproximación más cercana es lo suficientemente pequeña (indicada por círculos en la Figura 12). Se determina la más cercana de estas posibles colisiones, si la hubiera. Luego, el personaje gira para evitar el lugar de la colisión prevista. Girará lateralmente para alejarse del

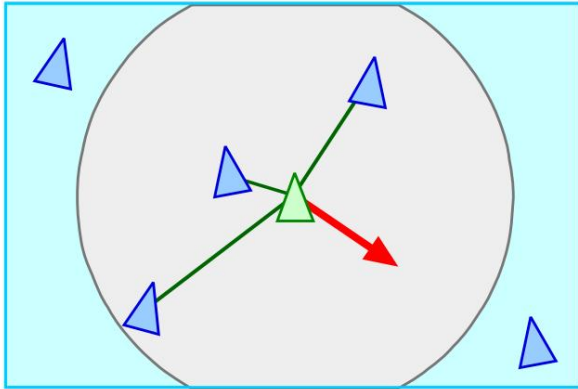


Figura 14: separación

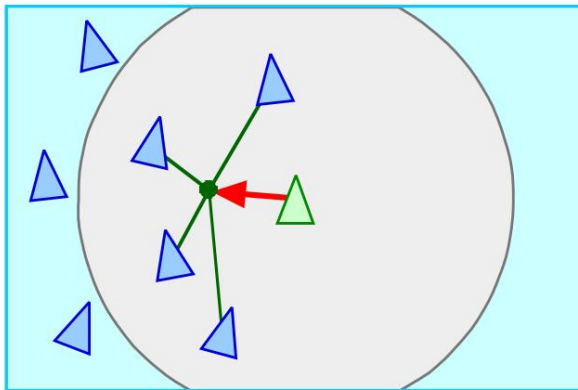


Figura 15: cohesión

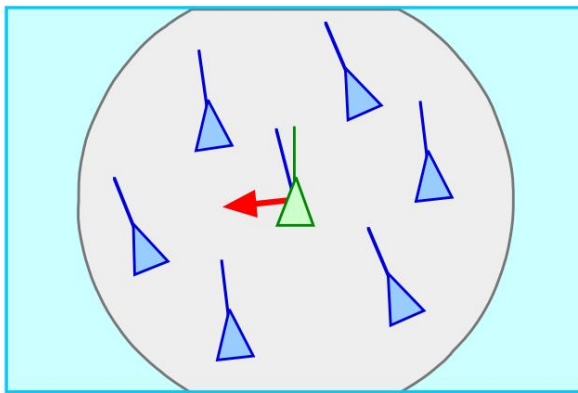


Figura 16: alineación

posible colisión. También acelerará hacia adelante o desacelerará hacia atrás para llegar al sitio indicado antes o después de la colisión prevista. En la Figura 12, el personaje que se acerca por la derecha decide reducir la velocidad y girar hacia la izquierda, mientras que el otro personaje acelerará y girará hacia la izquierda.

Los siguientes tres comportamientos directivos: separación, cohesión y alineación, se relacionan con grupos de personajes. En cada caso, el comportamiento de dirección determina cómo reacciona un personaje ante otros

personajes en su vecindario local. Se ignoran los personajes fuera del vecindario local. Como se muestra en la Figura 13, la vecindad se especifica mediante una distancia que define cuándo dos personajes están "cerca" y un ángulo que define el "campo de visión" perceptivo del personaje.

El comportamiento de dirección de separación le da al personaje la capacidad de mantener una cierta distancia de separación de otros cercanos. Esto se puede utilizar para evitar que los personajes se amontonen. Para calcular la orientación de la separación, primero se realiza una búsqueda para encontrar otros caracteres dentro de la vecindad especificada. Esta podría ser una búsqueda exhaustiva de todos los personajes en el mundo simulado, o podría utilizar algún tipo de partición espacial o esquema de almacenamiento en caché para limitar la búsqueda a caracteres locales. Para cada personaje cercano, se calcula una fuerza repulsiva restando las posiciones de nuestro personaje y el personaje cercano, normalizando y luego aplicando una ponderación $1/r$. (Es decir, el vector de desplazamiento de posición se escala en $1/r^2$.) Tenga en cuenta que $1/r$ es sólo una configuración que ha funcionado bien, no un valor fundamental. Estas fuerzas repulsivas de cada personaje cercano se suman para producir la fuerza de dirección general. Ver Figura 14.

El comportamiento de dirección de cohesión le da al personaje la capacidad de cohesionarse (acercarse y formar un grupo con) otros personajes cercanos. Véase la Figura 15. La dirección para la cohesión se puede calcular mediante

encontrar todos los personajes en el vecindario local (como se describió anteriormente para la separación), calcular la "posición promedio" (o "centro de gravedad") de los personajes cercanos. La fuerza de dirección se puede aplicar en la dirección de esa "posición promedio" (restando la posición de nuestro personaje de la posición promedio, como en el modelo boids original), o se puede usar como objetivo para buscar el comportamiento de dirección.

El comportamiento de dirección de alineación le da a un personaje la capacidad de alinearse con (es decir, dirigirse en la misma dirección y/o velocidad que) otros personajes cercanos, como se muestra en la Figura 16. La dirección para la alineación se puede calcular buscando todos los personajes en el mapa local. vecindad (como se describió

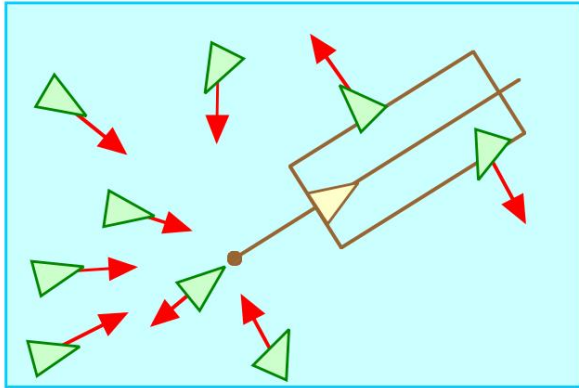


Figura 17: líder siguiendo

anteriormente para la separación), promediando juntos la velocidad (o alternativamente, el vector unitario de avance) de los caracteres cercanos. Este promedio es la "velocidad deseada", por lo que el vector de dirección es la diferencia entre el promedio y la velocidad actual de nuestro personaje (o alternativamente, su vector unitario de avance). Esta dirección tenderá a girar a nuestro personaje para que esté alineado con sus vecinos.

Comportamiento de bandada: además de otras aplicaciones, los comportamientos de separación, cohesión y alineación se pueden combinar para producir el modelo boids de bandadas, rebaños y escuelas [Reynolds 87] (ver también [Tu 94], [Tu 96] y [Hodgins 94]). En algunas aplicaciones, es suficiente simplemente sumar los tres

vectores de fuerza de dirección para producir una única dirección combinada para flocado (consulte Combinación de comportamientos a continuación). Sin embargo, para un mejor control es útil normalizar primero los tres componentes de la dirección y luego escalarlos mediante tres factores de ponderación antes de sumarlos. Como resultado, el comportamiento del flocado boid se especifica mediante nueve parámetros numéricos: un peso (para combinar), una distancia y un ángulo (para definir la vecindad, ver Figura 13) para cada uno de los tres comportamientos componentes.

El comportamiento de seguimiento del líder hace que uno o más personajes sigan a otro personaje en movimiento designado como líder. Generalmente los seguidores quieren permanecer cerca del líder, sin abarrotar al líder, y teniendo cuidado de mantenerse fuera del camino del líder (en caso de que se encuentren frente al líder). Además, si hay más de un seguidor, quieren evitar chocar entre sí. La implementación del seguimiento del líder depende de la llegada.

comportamiento (ver arriba) un deseo de avanzar hacia un punto, desacelerando a medida que se acerca. El objetivo de llegada está un punto ligeramente desviado del líder. (La distancia de desplazamiento puede aumentar opcionalmente con la velocidad). Si un seguidor se encuentra en una región rectangular frente al líder, se desviará lateralmente del camino del líder antes de reanudar el comportamiento de llegada. Además, los seguidores utilizan un comportamiento de separación para evitar que se amontonen entre sí. Ver Figura 17.

Finalmente, aquí hay breves esbozos de algunos otros comportamientos de dirección que encajan en la misma categoría general que los descritos con más detalle anteriormente. El comportamiento de dirección interpuesta intenta colocar su personaje entre otros dos personajes en movimiento, por ejemplo, un jugador de fútbol que intenta bloquear un pase entre dos miembros del equipo contrario. El enfoque general es similar a la persecución descrita anteriormente: predecir la posición futura de los otros dos personajes, determinar un punto objetivo interpolando entre las posiciones futuras y utilizar la búsqueda para dirigirse hacia el punto objetivo. En relación con el seguimiento y la persecución del líder, podríamos seguir a nuestra presa acercándonos y luego usando la alineación para igualar su velocidad y rumbo. La llegada

El comportamiento descrito anteriormente puede considerarse una limitación de la posición y la velocidad. Esto se puede ampliar para restringir simultáneamente la orientación (para producir un acoplamiento), o una velocidad distinta de cero, y/o para cumplir con estas restricciones en un momento dado. Ocultar el comportamiento implica identificar un objetivo

ubicación que está en el lado opuesto de un obstáculo del oponente y dirigirse hacia él usando la búsqueda.

Combinando comportamientos

Los comportamientos de dirección individuales descritos anteriormente sirven como bloques de construcción para patrones de comportamiento más complejos. Son componentes de una estructura más amplia, como las notas de una melodía o las palabras de una historia. Para crear comportamientos interesantes y realistas, necesitamos seleccionar y combinar estos componentes individuales. A menos que exista un personaje autónomo en un mundo muy simple, rara vez tendría sentido que el personaje ejecutara continuamente un único comportamiento de dirección.

La combinación de comportamientos puede ocurrir de dos maneras. Un personaje puede cambiar secuencialmente entre modos de comportamiento a medida que cambian las circunstancias en su mundo. Por ejemplo, imagina un caribú pastando en un prado cuando de repente siente que se acercan lobos. Este evento desencadena un cambio de comportamiento discreto. Todos los pensamientos sobre el pastoreo se olvidan cuando la manada de caribúes se da vuelta para huir de los depredadores. No hay tendencia a mezclar estos comportamientos: un caribú no disminuirá la velocidad mientras huye de un lobo para agarrar otro bocado de comida. Estos cambios discretos del estado conductual tienen lugar en el nivel de selección de acción, la parte superior de la jerarquía conductual de tres niveles analizada en la Introducción. Hay una extensa discusión sobre la selección de acciones en [Tu 94], [Tu 96] y en [Blumberg 94].

Por otro lado, algunos tipos de comportamientos comúnmente se combinan y actúan efectivamente en paralelo. Por ejemplo, cuando el caribú huye a través del bosque, combina la evasión y la evitación de obstáculos para permitirle escapar de los lobos mientras esquivo los árboles. Un caribú no puede darse el lujo de ignorar ninguno de los componentes del comportamiento; siempre debe moverse en una dirección que lo aleje del lobo y evite colisiones con los árboles. Esta combinación de comportamientos se produce en el nivel directivo medio de la jerarquía conductual.

La combinación de comportamientos de dirección se puede lograr de varias maneras. La más sencilla es simplemente calcular cada uno de los comportamientos de dirección de los componentes y sumarlos, posiblemente con un factor de ponderación para cada uno de ellos. (Tenga en cuenta que los vectores de dirección son especialmente fáciles de combinar; otros tipos de comportamientos que producen otros tipos de valores (por ejemplo, comportamientos conversacionales) podrían ser mucho más difíciles de combinar.) Esta combinación lineal simple a menudo funciona bien, pero tiene al menos dos desventajas: no es el enfoque más eficiente desde el punto de vista computacional y, a pesar de ajustar las ponderaciones, los comportamientos de los componentes pueden anularse entre sí en momentos inoportunos.

La carga de cálculo se puede reducir observando que el impulso de un personaje sirve para aplicar un filtro de paso bajo a los cambios en la fuerza de dirección. Entonces, en lugar de calcular varios componentes de la dirección en cada paso de la simulación y promediarlos juntos, podríamos seleccionar un componente de la dirección para calcular y aplicar cada cuadro, y depender del impulso (y tal vez de alguna amortiguación explícita de la aceleración) para combinarlos.

El problema de que los componentes se cancelen entre sí se puede abordar asignando una prioridad a los componentes. (Por ejemplo: la primera prioridad es evitar obstáculos, la segunda es evadir...) El controlador de dirección primero verifica si la evitación de obstáculos devuelve un valor distinto de cero (lo que indica una posible colisión); de ser así, lo utiliza. De lo contrario, pasa al comportamiento de segunda prioridad, y así sucesivamente.

Un híbrido de estas técnicas que el autor ha encontrado útil es el "dithering priorizado": con cierta probabilidad se evalúa el comportamiento de primera prioridad, y si devuelve un valor distinto de cero (no nulo), se utilizará. De lo contrario (si el comportamiento devuelve cero o se omitió debido a la selección aleatoria), se considera el comportamiento de segunda prioridad, y así sucesivamente.

En [Reynolds 87] se utilizó un esquema de combinación llamado "asignación de aceleración priorizada" con el modelo de flocking de boids. La idea básica era que, al ajustar su magnitud, los comportamientos de mayor prioridad podrían decidir si dejar o no alguna fuerza de dirección para que los comportamientos de menor prioridad los utilicen. En el transcurso de varias reimplementaciones de boids a lo largo de los años, una simple combinación lineal de los comportamientos de los componentes ha demostrado ser suficiente. Al combinar la agrupación con otros comportamientos, como evitar obstáculos, se han utilizado con éxito tanto la suma simple como el difuminado priorizado.

Conclusiones

Este artículo definió el "carácter autónomo" en términos de agentes autónomos y acción de improvisación. Presentó una descomposición de la tarea de construir comportamientos de movimiento para personajes autónomos en una jerarquía de tres niveles: selección de acciones, dirección y locomoción. Ha definido una implementación mínima del nivel de locomoción en términos de un "modelo de vehículo simple". Luego presentó una colección de comportamientos de dirección simples y comunes. (Incluyendo: buscar, huir, persecución, evasión, persecución compensada, llegada, evitación de obstáculos, deambular, seguimiento de camino, seguimiento de pared, contención, seguimiento de campo de flujo, evitación de colisiones no alineadas, separación, cohesión, alineación, agrupamiento y seguimiento del líder). Finalmente, se han descrito algunas técnicas para combinar estos comportamientos de dirección simples.

Expresiones de gratitud

Las técnicas descritas en este artículo se han desarrollado durante los últimos doce años, para muchos proyectos diferentes, en varias empresas. Deseo agradecer la útil cooperación de todas las empresas y compañeros de trabajo involucrados. Específicamente deseo agradecer a las siguientes personas por su apoyo gerencial y colaboración técnica. En Sony Computer Entertainment America: Phil Harrison, John Phua, Attila Vass, Gabor Nagy, Sky Chang y Tom Harper. En DreamWorks Feature Animation: Dylan Kohler, Bart Gawboy, Matt Arrott, Lance Williams, Saty Raghavachary y Mike Ullner. En Silicon Studio de SGI: Bob Brown, Leo Blume, Roy Hashimoto y, especialmente, mi colega de animación conductual Xiaoyuan Tu. En Electronic Arts: Luc Barthelet, Steve Crane, Kelly Pope, Steve Sims y Frank Giraffe. En la División de Gráficos Simbólicos: Tom McMahon, Andy Kopra, Larry Malone y Michael Wahrman. Finalmente, un cariñoso agradecimiento a mi esposa Lisa y a nuestros hijos Eric y Dana. Antes de que se vuelvan completamente autónomos, espero guiar a los niños en la dirección correcta. ¡Ciertamente ya son personajes reales!

Referencias

- Arkin, Ronald (1987) "Navegación basada en esquemas motores para un robot móvil: un enfoque de programación por Behavior", Actas de la Conferencia IEEE sobre Robótica y Automatización, páginas 264-271.
- Arkin, Ronald (1989) "Navegación por robot móvil basada en esquemas motores", Revista Internacional de Investigación en Robótica, 8(4) páginas 92-112.
- Arkin, Ronald (1992) "Navegación robótica basada en el comportamiento en dominios extendidos", Journal of Adaptive Behavior, 1(2) páginas 201-225.

Bates, José; Leal, Bryan; Reilly, Scott (1992) "Una arquitectura para la acción, la emoción y el comportamiento social", Actas del Cuarto Taller Europeo sobre Modelado de Agentes Autónomos en un Mundo Multiagente, S. Martino al Camino, Italia. <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/papers/CMU-CS-92-144.ps>

Cerveza, RD (1990). Inteligencia como comportamiento adaptativo: experimentos en neuroetología computacional. Nuevo York: Prensa académica. Ver también: <http://yuggoth.ces.cwru.edu/johng/robopaper/robopaper.html>

Blumberg, Bruce y Galyean, Tinsley (1995) Dirección multinivel de criaturas autónomas para entornos virtuales en tiempo real, Actas de SIGGRAPH 95, en Computer Graphics Proceedings, Serie de conferencias anuales, ACM SIGGRAPH, páginas 47-54. <http://bruce.www.media.mit.edu/people/bruce/Siggraph95.final.ps>

Blumberg, Bruce (1994) "Acción-Selección en Hamsterdam: Lecciones de Etología" en las Actas de la Tercera Conferencia Internacional sobre la Simulación del Comportamiento Adaptativo (SAB94), D. Cliff, P. Husbands, JA Meyer y S. Wilson, Editores, MIT Press, Cambridge, Massachusetts, páginas 108-117

Braitenberg, Valentino (1984) Vehículos: experimentos en psicología sintética, The MIT Press, Cambridge, MA.

Brooks, Rodney A. (1985) "Un robusto sistema de control en capas para un robot móvil", IEEE Journal of Robotics y Automation 2(1), marzo de 1986, págs. 14-23; también MIT AI Memo 864, septiembre de 1985. Ver: <http://www.ai.mit.edu/people/brooks/papers/AIM-864.pdf>

Bruderlin, Armin y Calvert, Tom (1989) "Animación dinámica y dirigida a objetivos de la marcha humana", ACM SIGGRAPH'89, Actas, vol. 23, págs. 233-242.

Cliff, Dave y Miller, Geoffrey (1996) "Co-Evolución de la persecución y la evasión II: métodos y resultados de simulación", De los animales a los animats 4: Actas de la Cuarta Conferencia Internacional sobre Simulación del Comportamiento Adaptativo (SAB96), Maes, Mataric, editores de Meyer, Pollack y Wilson, ISBN 0-262-63178-4, MIT Press. <http://www.cogs.susx.ac.uk/users/davec/pe.html>

Costa, Mónica; Feijó, Bruno; y Schwabe, Daniel (1990) Agentes reactivos en animación conductual, Anais do SIBGRAPI 95, editores Lotufo y Mascarenhas, São Carlos, SP, páginas 159-165. <http://www.icad.puc-rio.br/~monica/pubs.htm>

Cremer, James; Kearney, J.; Willemsen, P. (1996) "A Directable Vehicle Behavior Model for Virtual Driving Environments", Actas de la Conferencia de 1996 sobre IA, simulación y planificación en sistemas de alta autonomía, La Jolla, CA. <http://www.cs.uiowa.edu/~cremer/papers/aisp-final.ps>

Egbert, Parris y Winkler, Scott (1996) "Movimiento de objetos sin colisiones utilizando campos vectoriales", IEEE Computer Graphics and Applications, 16(4), páginas 18-24. <http://www.computer.org/cga/cg1996/g4toc.htm>

Johnson, Michael (1994) WavesWorld: un banco de pruebas para personajes animados tridimensionales semiautónomos, tesis doctoral, MIT Media Lab. <http://wave.www.media.mit.edu/people/wave/PhDThesis/outline.html>

Hayes-Roth, Bárbara. y van Gent, R. (1996) "Improvisational Puppets, Actors, and Avatars", Actas de la Conferencia de Desarrolladores de Juegos de Computadora de 1996. Informe del laboratorio de sistemas de conocimiento de Stanford KSL-96-09. [archivo://www-ksl.stanford.edu/pub/KSL_Reports/KSL-96-09.ps](http://www-ksl.stanford.edu/pub/KSL_Reports/KSL-96-09.ps)

Hodgins, Jessica y Brogan, David (1994) "Robot Herds: Group Behaviors for Systems with Significant Dynamics", Actas del 4º Taller Internacional sobre Síntesis y Simulación de Sistemas Vivos (Artificial Life IV), editores de Brooks y Maes, MIT Press, Cambridge, MA, páginas 319-324. <http://www.cc.gatech.edu/gvu/animation/papers/alife.ps.gz>

Hodgins, Jessica; Wooten, W; Brogan, D; y O'Brien, J (1995) "Animating Human Athletics", Actas de SIGGRAPH 95, en Computer Graphics Proceedings, Serie de conferencias anuales, editor de Robert Cook, ACM SIGGRAPH, páginas 71-78. <http://www.cc.gatech.edu/gvu/animation/papers/sig95.ps.gz>

Hubbard, Philip (1996) "Aproximación de poliedros con esferas para la detección de colisiones en tiempo crítico", ACM Transacciones sobre gráficos, 15 (03), páginas 179-210. <http://www.acm.org/pubs/tog/hubbard96/>

Isaacs, Rufus (1965) Juegos diferenciales: una teoría matemática con aplicación a la guerra y la persecución, el control y la optimización, John Wiley and Sons, Nueva York.

Kahn, Kenneth (1979) "Creación de animación por computadora a partir de descripciones de historias", Informe técnico 540, MIT Laboratorio de IA, Cambridge, Massachusetts.

Maes, Pattie; Darrell, T.; Blumberg, B. (1995) "El sistema vivo: interacción de cuerpo completo con agentes autónomos", Conferencia de animación por computadora '95, IEEE Press, páginas 11-18.

Mataric, Maja (1993) "Diseño y comprensión del comportamiento grupal adaptativo", Adaptive Behavior 4(1), páginas 51-80. <http://www-robotics.usc.edu/~maja/> <http://www-robotics.usc.edu/~maja/publications/abj95.ps.gz>

Ohashi, Yoshikazu (1994) "Aproximaciones lineales rápidas de distancia euclidiana en dimensiones superiores", en Graphics Gems IV, editor de Paul Heckbert, Academic Press. Consulte <ftp://princeton.edu/pub/Graphics/GraphicsGems/GemsIV/>

Perlin, Ken (1985) "An Image Synthesizer", en SIGGRAPH '85 Proceedings, Computer Graphics 19(3), páginas 287-296. Ver <http://www.mrl.nyu.edu/perlin/doc/oscar.html>

Perlin, Ken (1995) "Animación receptiva en tiempo real con personalidad", IEEE Transactions on Visualization

y Computer Graphics, 1(1) páginas 5-15, ISSN 1077-2626.

Perlin, Ken y Goldberg, Athomas (1996) "Improv: A System for Scripting Interactive Actors in Virtual Worlds", Actas de SIGGRAPH 96, en Computer Graphics Proceedings, Serie de conferencias anuales, ACM SIGGRAPH, páginas 205-216. <http://www.mrl.nyu.edu/improv/> <http://www.mrl.nyu.edu/improv/sig-paper96/>

Pottinger, Dave (1999) "Movimiento de unidades coordinadas" e "Implementación del movimiento coordinado", Juego Developer Magazine, enero de 1999. Consulte:

http://www.gamasutra.com/features/game_design/19990122/movement_01.htm http://www.gamasutra.com/features/game_design/19990129/implementing_01.htm

www.gamasutra.com/features/game_design/19990129/implementing_01.htm

Raibert, Marc y Hodgins, Jessica (1991) "Animación de locomoción dinámica de piernas", Gráficos por computadora 25(4), Actas de SIGGRAPH 91, editor de Thomas Sederberg, ISSN 0097-8930, páginas 349-358.

Raibert, M., Hodgins, J., Ringrose, R., Playter, R., Borvansky, L., Campbell, L., Evans, D., Crane, C., Lamb, M. (1991) "On The Run", SIGGRAPH '91 Electronic Theatre, SIGGRAPH Video Review, número 72, <http://www.ai.mit.edu/projects/leglab/simulations/otr/otr.html>

Reese, Bjørn y Bryan, Stout (1999) "Finding a Pathfinder" aparecerá: Simposio de primavera AAAI 99 sobre Inteligencia artificial y juegos de ordenador. Véase también: <http://www.red.com/breese/navigation.html>

Resnick, Mitchel (1989) "LEGO, Logo, and Life", Actas del taller interdisciplinario sobre síntesis y simulación de sistemas vivos (ALife '87), Estudios SFI en ciencias de la complejidad, volumen 6, editor de Christopher Langton, Addison-Wesley, Redwood City, CA, EE. UU., páginas 397-406.

Resnick, Mitchel (1993) Kits de construcción de comportamiento, CACM, 36(7), <http://el.www.media.mit.edu:80/groups/el/Papers/mres/BCK/BCK.html>

Resnick, Mitchel et al. (1998) "La pecera virtual" <http://el.www.media.mit.edu/groups/el/projects/fishtank/>

Reynolds, CW (1987) Bandadas, rebaños y escuelas: un modelo de comportamiento distribuido, en Computer Graphics, 21 (4) (Actas de la conferencia SIGGRAPH '87) páginas 25-34. <http://www.red.com/cwr/boids.html>

Reynolds, CW (1988) Not Bumping Into Things, en las notas del curso SIGGRAPH 88 Desarrollos en Modelado basado en física, páginas G1-G13, publicado por ACM SIGGRAPH. <http://www.red.com/cwr/nobump/nobump.html>

Ridsdale, Gary (1987) "El aprendiz de director: animando figuras en un entorno restringido", Ph.D. Tesis, Facultad de Ciencias de la Computación, Universidad Simon Fraser.

Sims, Karl (1994) "Evolving Virtual Creatures", Actas de SIGGRAPH 94, Computer Graphics Proceedings, Serie de conferencias anuales, editor de Andrew Glassner, ACM SIGGRAPH, ISBN 0-89791-667-0, páginas 15-22. <ftp://think.com/users/karl/Welcome.html> <ftp://think.com/users/karl/siggraph94.ps>

Aún así, G Keith (1994) "Simulación de salida utilizando la realidad virtual: una perspectiva de simulación y diseño",

IMAS Seguridad contra Incendios en Buques.

Strassmann, Steve (1991) Desktop Theatre: generación automática de animación expresiva, tesis doctoral, MIT Laboratorio de medios <http://www.method.com/straz/straz-phd.pdf>

Thalmann, Daniel; Renault, Olivier y Magnenat-Thalmann Nadia (1990) "Un enfoque basado en la visión para la animación conductual", Journal of Visualization and Computer Animation, John Wiley & Sons, 1(1) páginas 18-21.

Travers, Michael (1989) "Animal Construction Kits", Actas del taller interdisciplinario sobre síntesis y simulación de sistemas vivos (ALife '87), Estudios SFI en las ciencias de la complejidad, volumen 6, editor de Christopher Langton, Addison-Wesley, Redwood City, CA, EE. UU., páginas 421-442. <http://lcs.www.media.mit.edu/people/mt/agar/agar.html>

Travers, Michael (1994) "LiveWorld: un kit de construcción para sistemas animados", Actas de ACM CHI'94 Conferencia sobre factores humanos en sistemas informáticos, páginas 37-38.

<http://mt.www.media.mit.edu/people/mt/papers/chi94/chi94.html>

Tu, Xiaoyuan y Terzopoulos, Demetri (1994) "Peces artificiales: física, locomoción, percepción, comportamiento", Actas de SIGGRAPH 94, Actas de gráficos por computadora, Serie de conferencias anuales, editor de Andrew Glassner, ACM SIGGRAPH, ISBN 0-89791-667-0, páginas 43-50. <http://www.dgp.toronto.edu/people/tu/papers/sig94.ps>

Tu, Xiaoyuan (1996) Animales artificiales para animación por computadora: biomecánica, locomoción, percepción y comportamiento, tesis doctoral, Departamento de Ciencias de la Computación, Universidad de Toronto. <http://www.dgp.toronto.edu/people/tu/thesis/thesis.html>

van de Panne, M., Fiume, E. y Vranesic, ZG, (1990) "Síntesis de movimiento reutilizable utilizando controladores de espacio-estado", Actas de SIGGRAPH '90, en Computer Graphics Proceedings, ACM SIGGRAPH, páginas 225-234. Ver <http://www.dgp.toronto.edu/people/van/papers.html>

Walter, W. Gray (1950) "Una imitación de la vida", Scientific American, 182(5), páginas 42-45. (ver también <http://gate.uwe.ac.uk:8002/IAS/gwonline.html> y <http://www.newscientist.com/ns/980725/robotman.html>)

Grey-Walter, W. (1953) El cerebro vivo. Gerald Duckworth y Co., Ltd.

Wiener, Norbert (1948) Cibernética o control y comunicación en el animal y la máquina.

Cambridge, Massachusetts: The Technology Press; Nueva York: John Wiley & Sons, Inc.

Wilhelms, Jane y Skinner, Robert (1990) Una "noción" para el control interactivo de la animación conductual, IEEE Computer Graphics and Applications, 10(3), páginas 14-22.

Zapata, R., Lepinay, P., Novales, C. y Deplanques, P. (1992) "Comportamientos reactivos de robots móviles rápidos en entornos no estructurados: control basado en sensores y redes neuronales" en From Animals to Animats 2: Actas de la Segunda Conferencia Internacional sobre Simulación del Comportamiento Adaptativo (SAB92), editores de Meyer, Roitblat y Wilson, MIT Press, Cambridge, Massachusetts, páginas 108 -115

Zeltzer, David (1983) Animación basada en el conocimiento, Actas del Taller sobre movimiento SIGGRAPH/SIGART, páginas 187-192.

Zeltzer, David (1990) "Simulación gráfica a nivel de tarea: abstracción, representación y control", en Making
Se mueven: mecánica, control y animación de figuras articuladas, N. Badler, B. Barsky y D. Zeltzer, editores, Morgan Kaufmann Publishers.