

Crowd Simulation

Tibor Schütz

Quelle: www.deinmeister.de

23. Januar 2004

Inhaltsverzeichnis

1	Crowd Simulation	1
1.1	Crowd Simulation, Artificial Life (AL) und Künstliche Intelligenz (KI)	1
1.2	Bedarf	2
2	Kontinuierliche Simulation	3
3	Diskrete Simulation	4
3.1	Zelluläre Automaten	4
3.1.1	Beispiel: Verkehrssimulation mit 1D-Automat	4
3.2	Partikel- und Multiagentensysteme	5
3.2.1	Vereinfachungen	5
3.3	Diskretisierung des Raumes	6
4	Verhaltensmodellierung	7
4.1	Anforderungen	7
4.2	Implementierung	7
4.3	Beispiel: <i>The Sims</i>	8
4.3.1	Haupteigenschaften	8
4.3.2	Grundalgorithmus	8
5	Ausstattung der Umgebung	9
5.1	Aktionspunkte	9
5.2	Anziehende und abstoßende Punkte	9
5.3	Interaktive Objekte	10
6	Kollisionserkennung	11
7	Wegsuche	12
7.1	Anforderungen	12
7.2	Implementierung	12
8	Indirekte Kommunikation zwischen Individuen	14
8.1	Fallbeispiel: Entgegenkommende Objekte an einer engen Passage	14

9 Herdenbildung	15
9.1 Implizit	15
9.2 Explizit	15
9.2.1 Ein Beispiel für die Regelmenge als Herdenmitglied	16
10 Gruppendynamik	17
11 Kontrolle der Simulation	18
12 Darstellung	19
13 Adressen	20

Kapitel 1

Crowd Simulation

1.1 Crowd Simulation, Artificial Life (AL) und Künstliche Intelligenz (KI)

Die Simulation eines aus einzelnen autonomen Individuen zusammengesetzten Systems gehört zum Kerngebiet der Simulation künstlichen Lebens (AL). Bei diesen Individuen, im Zusammenhang der Simulation oft auch als Agenten bezeichnet, kann es sich um Menschen, Tiere, Fahrzeuge oder auch generalisierte Gebilde wie Haushalte handeln. Charakteristisch für die Agenten ist ein Verhalten, das sich nicht allein durch Naturgesetze beschreiben läßt, sondern auch Methoden der KI erfordert.

Aufgrund des Fehlens allgemein anerkannt exakter Modelle zur Beschreibung solcher Systeme beschränkt sich die Überprüfung derartiger Simulationen meist nur darauf, ob sie im Rahmen der Erwartungen lauffähig ist und schlüssige Ergebnisse berechnet. Dafür ist in vielen Anwendungsbereichen die Echtzeitfähigkeit von großer Bedeutung. Die einer Crowd Simulation zugrunde liegenden Techniken und Regeln werden deshalb bevorzugt nach Effizienzkriterien ausgewählt. Von Vorteil ist, daß hierbei in vielen Fällen auf schon bewährte Algorithmen aus anderen Gebieten zurückgegriffen werden kann.

1.2 Bedarf

Die Simulation vieler zusammen agierender Individuen besitzt ein weitgefächertes Anwendungsspektrum:

- Die Nachbildung großer Menschenmengen und Tierherden in Filmen erlaubt die Realisierung anderweitig nur mit sehr hohem Aufwand an Statisten oder Tricktechnik darstellbarer Szenen. Dementsprechend sind hier Simulationen gefordert, die optisch möglichst wirklichkeitsnah erscheinen. Eine gute Zusammenarbeit zwischen der Simulation der Menge und der Simulation der einzelnen Agenten ist hierbei besonders wichtig.
- Bei Entwurf von Gebäuden und Anlagen läßt sich deren Benutzung im Alltag simulieren. Neben demonstrativen und ästhetischen Aspekten ist hier vor allem die Planung von Fluchtwegen das Haupteinsatzgebiet. Aber auch zur Komfortsteigerung durch Optimierung von Ein- und Ausgängen oder für die Auswahl geeigneter Baumaterialien bieten sich Simulationen an. Hauptkriterium für die Wahl der Simulation ist ein möglichst realistisches Verhalten der Menge.
- Ein benachbartes Einsatzgebiet ist die Verkehrsplanung und -steuerung, hauptsächlich zur Verbesserung des Verkehrsflusses bei möglichst geringem Zusatzaufwand. Zum Verständnis und Vermeidung von realen Effekten wie Verkehrsstockungen sind Simulationen ebenfalls im Einsatz. Zu Test- und Trainingszwecken können diese auch interaktive Eingriffe ermöglichen.
- Interaktive, echtzeitfähige Simulationen mit weitgehend realistischem Verhalten werden mitunter auch in Ausbildung und Training von Einsatzkräften wie Rettungsdienste, Polizei, Militär, etc. eingesetzt.
- Nichtspielercharaktere in Computerspielen lassen sich durch echtzeitfähige, zur Laufzeit interaktive Simulationen in großer Zahl einfach realisieren. Die weiteren Anforderungen und Einsatzgebiete entsprechen denen in Filmen.
- Von gänzlich anderem Charakter ist die Verwendung in der Forschung: Das Verständnis von soziologischen und verhaltenspsychologischen Effekten läßt sich in eine Simulation umsetzen und somit getestet und erweitert werden. Idealerweise sollte sich die Simulation dazu möglichst leicht in ihren Regeln und Parametern ändern lassen.

Kapitel 2

Kontinuierliche Simulation

Zur Simulation von Gasen und Flüssigkeiten mittels (Differential-)Gleichungssystemen existieren schon länger ausgereifte und effiziente Verfahren mit einem soliden theoretischen Unterbau. Es liegt also nahe, diese auch zur Crowd Simulation einzusetzen. Nichtphysikalische Effekte müssen dabei durch Potentiale, Energien oder Felder nachgebildet werden.

Alle in der Praxis verwendeten Verfahren arbeiten auf einem in diskrete Zellen aufgeteilten Raum. Der Zustand einer Zelle gibt die Anzahldichte der darauf befindlichen Individuen und ggf. weitere Eigenschaften wie deren mittlere Geschwindigkeit wieder. In reinen Strömungssimulationen wird nur die unmittelbare Umgebung einer jeden Zelle in den dazugehörigen Berechnungen verwendet.

Die Stärken dieses kontinuierlichen Modells liegen in seiner umfangreichen Analysierbarkeit, was eine leistungsfähige Datenaufbereitung in Hinsicht auf Extremstellen, Mittelwerte, Schwankungen, etc. ermöglicht. Sich einstellende Gleichgewichte lassen sich berechnen, zudem läßt sich die Genauigkeit der Berechnungen (aber nicht des zugrundeliegenden Modells) exakt kontrollieren.

Diese Art der Modellierung eignet sich allerdings nur für wenige Einsatzgebiete, da sie das Konzept unabhängiger, verschiedener Individuen vollkommen ignoriert.

Kapitel 3

Diskrete Simulation

3.1 Zelluläre Automaten

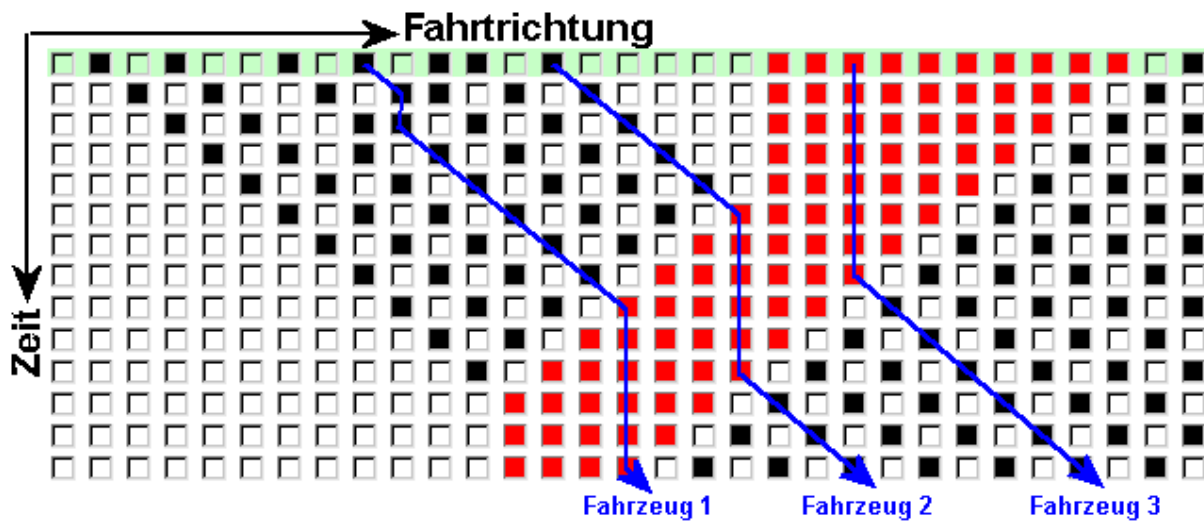
Spätestens seit Conways *Game of Life* gehören zelluläre Automaten zu den Standardverfahren in der Imitation künstlichen Lebens. Auch hier wird ein in Zellen aufgeteilter Raum verwendet, der Zustand einer Zelle wird in jedem Zeitschritt mithilfe der benachbarten Zellen, also lokal, berechnet. Der Zellenzustand kann entweder wie im klassischen zellulären Automat nur aus diskreten Werten bestehen oder auch um kontinuierliche Werte und globale Parameter erweitert sein. Bei geeigneter Wahl der Zustände ist sogar die Nachbildung einzelner Individuen möglich. Komplexe Verhaltensweisen lassen sich allerdings nur mit übermäßig hohem Aufwand realisieren.

3.1.1 Beispiel: Verkehrssimulation mit 1D-Automat

Mit folgenden Regeln läßt sich eine einfache Verkehrssimulation durchführen:

- Ist ein Feld belegt und das nächstliegende frei, so leere das Feld
- Ist ein Feld frei und das vorhergehende belegt, so belege das Feld
- In allen anderen Fällen bleibt der Zustand unverändert

Darstellung eines Simulationslaufs:



Jede Zeile entspricht einem Zeitschritt, der grün hervorgehobene Bereich stellt den Ausgangszustand dar. Schwarze Felder entsprechen einzelnen Fahrzeugen. Zum besseren Verständnis wurden die simulierte Bewegung dreier einzelner Fahrzeuge sowie der Bereich einer sich auflösenden Verkehrsstockung nachträglich eingezeichnet. Die momentane Geschwindigkeit eines Fahrzeugs läßt sich an der horizontalen Komponente der blauen Linien ablesen.

3.2 Partikel- und Multiagentensysteme

Am flexibelsten sind Verfahren, die sich vom Konzept des Raumes lösen und stattdessen direkt einzelne Agenten simulieren. Diese können als Partikelsystem oder auch als ein System aus unabhängig voneinander berechneten Agenten (Multi Agent System) realisiert werden. Prinzipiell sind beide Verfahren gleich leistungsfähig. In der Praxis sind Partikelsysteme etwas effizienter zu implementieren, Agentensysteme dafür leichter um individuelle Eigenschaften und Funktionen erweiterbar.

3.2.1 Vereinfachungen

Der Raum auf dem sich die Agenten befinden ist zuerst einmal beliebig wählbar. Dies erlaubt reichhaltige Verhaltensweisen, vergrößert aber auch den Rechenaufwand für viele Aktionen enorm. Um dennoch viele Agenten gleichzeitig simulieren zu können, bietet es sich an, anstelle eines normalen Partikelsystems eine Partikelwarteschlange zu verwenden. Bei diesem werden nur noch diejenigen Agenten bearbeitet, bei denen eine bedeutende Zustandsänderung eintritt. Dazu ist allerdings ein Modell erforderlich, bei dem einzelne Agenten nicht in jedem Zeitschritt unbedingt aktualisiert werden müssen.

Eine beliebte Technik ist es, zusätzlich den Raum wieder in diskrete Zellen aufzuteilen und die dadurch entstehenden Nachbarschaftseigenschaften zur Verringerung des Rechenaufwands heranzuziehen. Mitunter wird ein auf diesem Raum arbeitender zellulärer Automat zur dynamischen Berechnung ortsgebundener Eigenschaften mit einem Partikel- oder Multiagentensystem kombiniert.

3.3 Diskretisierung des Raumes

Bei der Aufteilung des Raumes sind reguläre Gitter wegen der einfacheren Verwendung vorteilhaft. Verbreitet sind vor allem leicht in Ortskoordinaten umrechenbare quadratische und kubische Gitter, da sie für frei bewegliche Objekte ideal sind. Im zweidimensionalen Fall sind auch hexagonale Unterteilungen sinnvoll: Diese kommen nicht nur der Kreisform am nächsten, sie haben darüber hinaus gegenüber rechtwinkligen Gittern den Vorteil, daß die Mittelpunkte aller umgebenden Felder gleich weit entfernt sind.

Kapitel 4

Verhaltensmodellierung

4.1 Anforderungen

Kernstück der realistischen Simulation eines Individuums ist die Nachbildung seiner Aktionen und Reaktionen. Diese basieren wiederum auf der Imitation von individuellen Zielen, Wahrnehmungen und Gefühlen. Für langfristige Aktionen kann es erforderlich werden, Wissen, Gedächtnis oder gar Lerneffekte mit zu berücksichtigen. Zusätzlich zu den selbstständig berechneten Verhaltensweisen ist es in vielen Fällen wünschenswert, das Verhalten mit Skripten zu erweitern oder gezielt von außerhalb interaktiv zu steuern.

Die Modellierung des Verhaltens kann auch Daten für die Darstellung liefern: Man denke nur an die Darstellung einer Begrüßung, die Wirkung verschiedener Gesichtszüge anhand der gerade vorherrschenden Emotionen und akustischen Informationen wie verschiedenen laute Schritte, Ausrufe oder Atemgeräusche.

4.2 Implementierung

Im wesentlichen gibt es drei verbreitete Verfahren zur Aktualisierung des Zustands eines Agenten:

- Endliche Automaten, deren Zustandsübergänge das Verhalten simulieren. Diese können um Erweiterungen wie Strafautomaten mit bewerteten Übergangskanten, variable Regelmengen oder nicht boolesche Zustände versehen sein.
- Berechnung des neuen Zustandes als numerische Funktion des Ausgangszustands. Ideal für kontinuierliche Zustände, kann aber bei gegenseitigen Abhängigkeiten und ungeschickter Wahl der Funktionen zu unerwünschten Schwingungen zwischen Zuständen oder Singularitäten führen.

- Der Einsatz von Fuzzy Logic kombiniert die leicht erstellbare Regelbasis von Automaten mit den kontinuierlichen Zustandsübergängen von numerischen Funktionen

Aus Gründen der einfacheren Entwicklung und besseren Effizienz bieten sich weitere Modifikationen an:

- Aktionswarteschlangen können verwendet werden, um nur die jeweils wichtigsten Aktionen zu berechnen. Reaktionen haben hierbei stets die höchste Prioritäten und können mit einem Verfalldatum versehen sein.
- Für die Imitation von Wissen (z.B. die Lage eines bestimmten Gegenstands) bietet es sich an, die dazugehörigen Informationen global für alle Agenten schon fertig aufbereitet in einer Karte abzuspeichern.
- Die Einbindung eines Zufallsgenerators kann bei gleicher Regelmenge die Zahl der Verhaltensweisen vervielfachen.

4.3 Beispiel: *The Sims*

Die einzelnen Personen in der Alltagssimulation *The Sims* baut im Wesentlichen auf folgendem Schema auf:

4.3.1 Haupteigenschaften

- 5 feste Wesenszüge
- 6 entwickelbare Grundfähigkeiten
- 8 Stimmungen und Bedürfnisse
- Liste evtl. bestehender Beziehungen

4.3.2 Grundalgorithmus

1. Wähle zu verbessernden Zustand aus der Menge der Eigenschaften der Person aus
2. Gehe zu Objekt oder anderer Person, welche diesen Zustand verbessern kann
3. Führe dazugehörige Aktion aus
4. Aktualisiere die durch die Aktion beeinflussten Zustände

Kapitel 5

Ausstattung der Umgebung

In der realen Welt stellt die Interaktion mit der Umgebung als auch mit anderen Individuen ein Hauptmerkmal des Lebens dar. Dies läßt sich modellieren, indem man einzelne Bereiche und Individuen um folgende Merkmale bereichert:

5.1 Aktionspunkte

Unter Aktionspunkten versteht man Objekte, die gezielt bestimmte Verhaltensregeln ansprechen. Vorhandene Ziele eines Agenten sind oftmals an spezielle Aktionspunkte gekoppelt. Je nach Bedeutung des Aktionspunktes kann dieser auch über Begrenzungen verfügen und, z.B. für einen Fahrstuhl, nur eine bestimmte Anzahldichte/Traglast gleichzeitig zulassen, so daß er nicht mehr allen Agenten gleichzeitig zur Verfügung stehen kann.

5.2 Anziehende und abstoßende Punkte

Außergewöhnlich interessante Objekte als auch solche, mit denen ein Kontakt wenig begehrenswert erscheint, lassen sich mit Hilfe attraktiver und abstoßender Potentiale erstellen. Üblich sind hierfür mit zunehmender Distanz asymptotisch monoton abfallende Funktionen, die als Maximum einen endlichen oder unendlichen Wert annehmen. Unendliche Werte stellen abstoßende Punkte, die nie erreicht werden sollen, oder Attraktoren, die wie ein Schwarzes Loch nicht mehr verlassen werden können, dar.

Die Gewichtung des Potentials kann je nach Agent variieren und sich von Verhaltensregeln gesteuert bei diesen dynamisch verändern - etwa, um nachlassendes Interesse an einem Gegenstand zu modellieren. Ein und dieselbe Funktion kann dadurch sowohl ein starkes als auch schwaches, anziehendes und abstoßendes Objekt zugleich simulieren.

5.3 Interaktive Objekte

Tritt ein Agent mit einem Objekt in Kontakt, so kann sich neben dem Zustand des Agenten auch der des betreffenden Objekts ändern. Dies betrifft nicht nur die Interaktion zweier Individuen, sondern auch Objekte, die nicht beliebig oft aufgesucht werden können; aufsammlbare Gegenstände als auch jegliche Art von simulierten Automaten wie z.B. ein Fahrstuhl.

Es bietet sich an, nicht explizit zwischen passiven Objekten und Agenten zu unterscheiden. Dies vereinfacht den Systementwurf enorm, kann aber je nach Modellwelt den Rechenaufwand enorm steigern, da dann auch unveränderte passive Objekte in jedem Zeitschritt bearbeitet werden.

Kapitel 6

Kollisionserkennung

Damit eine Interaktion zwischen Objekten stattfinden kann muß zuerst einmal überprüft werden, ob sich diese an der gleichen Position befinden. Der naive Ansatz, für jedes Objekt die Distanz zu allen anderen Objekten zu prüfen führt zu quadratischem Gesamtaufwand, was nur für gering besiedelte Welten sinnvoll ist.

Im eindimensionalen Fall bietet es sich an, die Objekte anhand ihrer Position in einer Liste zu sortieren. Die Ermittlung der benachbarten Objekte eines Objektes ist damit in konstanter Zeit möglich. Unter der realistischen Annahme, daß sich die Objekte nicht beliebig schnell bewegen, läßt sich auch die jeweilige Listenposition in konstanter Zeit aktualisieren. Dies führt zu linearem Gesamtaufwand.

Ansonsten kann man die Welt wieder wie schon bei den zellulären Automaten in Zellen aufteilen. In jeder Zelle werden Verweise auf die darin befindlichen Objekte gespeichert, so daß für jeden Agent nur noch die Objekte in den benachbarten Zellen geprüft werden müssen. Für gleichverteilte Objekte ergibt sich ein linearer Gesamtaufwand. Wenn einzelne Zellen dicht besiedelt sind steigert sich dieser enorm, die Zellengröße sollte demnach so gewählt werden, daß stets nur ein kleiner Teil aller Objekte sich darin befinden wird.

Dieses Problem umgeht man, wenn jede Zelle nur ein einziges Objekt enthalten kann. Die dazu nötigen geringe Zellengröße führt allerdings zu enormer Zellenanzahl mit entsprechendem Speicherbedarf. Demnach empfiehlt sich dieses Verfahren vorwiegend für sehr dicht besiedelte Simulationen.

Baumstrukturen wie Quadrees oder Octrees gehen dagegen stets effizient mit dem Speicherplatz um. Eine Änderung der Objektposition ist dafür mitunter vergleichsweise kostspielig; der Gesamtaufwand kann in ungünstigen Fällen bis zu quadratischer Ordnung anwachsen.

Kapitel 7

Wegsuche

7.1 Anforderungen

Ziel der Wegsuche ist es, einen möglichst optimalen Weg von einem Startpunkt zu einem Zielpunkt zu finden. Die Optimierungskriterien wie benötigte Zeit und Energie können sich bei verschiedenen Suchanfragen ändern, auch soll bei der Wegsuche das begangene Terrain auf Kriterien wie Höhenunterschiede und verschiedene Untergründe berücksichtigt werden. Idealerweise erfolgt die Wegsuche derart, daß unerwünschte Kollisionen mit anderen aktiven Objekten vermieden werden.

7.2 Implementierung

Gängige auf Zellen arbeitende Verfahren sind der A*- und der Duftstoff-Algorithmus. Beide beruhen auf der Wahl des lokalen Optimums. Während der A*-Algorithmus jedoch mit einer Schätzung des restlichen Weges arbeitet basiert der aufwendigere Duftstoff-Algorithmus auf der Rückwärtsverfolgung eines sich vom Zielpunkt ausbreitenden simulierten Lockstoffes.

Ein anderer Ansatz simuliert die Wahrnehmung der Agenten. Hierbei wird innerhalb von einem bestimmten Suchradius diejenige Richtung gewählt, die am günstigsten dem Ziel entgegen zu führen scheint. Je nach Gelände und Suchradius kann der daraus resultierende Weg beliebig stark vom optimalen Weg abweichen oder überhaupt nicht gefunden werden. Dafür gibt er die Orientierung in freiem und unbekanntem Gelände gut wieder.

Der Aktionsradius, die Sichtweite als auch die Geschwindigkeit kann sich an das Gelände anpassen. Insbesondere beim vorhergehenden Ansatz ist es zudem sinnvoll, Geschwindigkeit und Sichtradius daran anzupassen, wie stark die gewählte Richtung tatsächlich zu dem Ziel weist.

Um den Aufwand der Wegsuche zu minimieren empfiehlt es sich, vorberechnete Infor-

mationen in der Karte zu speichern (statische KI). Dazu gehört neben der Markierung von Sackgassen auch die Speicherung kompletter Wegenetze, so daß die Wegsuche nur noch die Strecke zwischen Start- und Zielpunkt sowie dazugehörigen Punkten auf dem schon gespeicherten Wegenetz berücksichtigen muß. Das Wegstück auf dem vorberechneten Netz kann entweder mit bekannten Graphenalgorithmien wie z.B. von Dijkstra ermittelt werden oder als schon fertige Liste auf der Karte verfügbar sein.

In sehr einfach strukturierten Umgebungen genügen Kurven niederer Ordnung zwischen Ausgangspunkt und Ziel sowie Schnittpunktberechnungen mit potentiellen Hindernissen für die Wegsuche. Kollisionen werden je nach Anforderung an die Simulation durch variieren der Kurve oder zufällige Wahl einer neuen Richtung gelöst.

Kapitel 8

Indirekte Kommunikation zwischen Individuen

Die Nachbildung von vorausschauendem Handeln gehört zu den aufwendigsten Bereichen der KI, selbst wenn alle Daten der anderen Objekte direkt einsehbar sind und somit auf die Nachbildung der individuellen Wahrnehmung weitgehend verzichtet werden kann. Einzig praktikabel im Umfeld der Simulation großer Mengen ist derzeit die Interpolation und Extrapolation der Daten anderer Objekte in Abhängigkeit von der Zeit für eine bessere Kollisionsvermeidung oder die Anpassung individueller Ziele. In den meisten Fällen wird man daher Intelligenz nur auf einer niedrigeren Stufe imitieren.

8.1 Fallbeispiel: Entgegenkommende Objekte an einer engen Passage

Schon einfache Situationen wie enge Durchgänge erfordern einen umfangreichen Aufwand zu deren korrekter Erkennung. Der triviale Ansatz, bei sich gegenseitig blockierenden Agenten die Auflösung des Konflikts dem Zufall oder dem allgemeinen Verhalten (z.B. *der Klügere gibt nach* oder *der Stärkere gewinnt*) zu überlassen, eignet sich nur für die Simulation primitiver Lebewesen.

Mit einigen Kompromissen läßt sich das Problem aber durch den Einsatz von Aktionspunkten lösen: Entweder man platziert an den Enden des Durchgangs jeweils Aktionspunkte, die die Funktion unsichtbarer Pförtner übernehmen, oder man deklariert den gesamten Durchgang als Gebiet mit einem auf ein einziges Objekt beschränktem Fassungsvermögen. Zu beachten ist allerdings, daß die Agenten den Durchgang weiterhin als solchen erkennen müssen und ggf. warten können.

Kapitel 9

Herdenbildung

Bisher ging es vorwiegend um die Simulation einzelner Individuen. Jedoch gehören auch gemeinsam handelnde Gruppen zum Alltag. Dies läßt sich in der Modellierung des Verhaltens mit berücksichtigen:

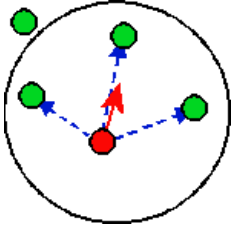
9.1 Implizit

Im einfachsten Fall benötigt man keine weiteren Regeln, sondern nur geeignete Ausgangszustände, um eine Herde zu simulieren. Gleiche Ziele und dieselbe Bewegungsrichtung genügen, damit unabhängige Agenten als Herde wahrgenommen werden. Problematisch ist allerdings die schwierige Kontrolle der Simulation, da schon geringe Unterschiede der Ausgangszustände und des Verhaltens zu völlig verschiedenen Ergebnissen führen können. Für interaktive Anwendungen ist dieser Ansatz deshalb nicht geeignet

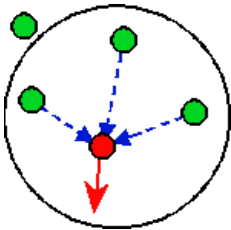
9.2 Explizit

Erweiter man den Zustandsvektor eines Agenten um die Zugehörigkeit zu einer Gruppe, sind Verhaltensregeln möglich, welche speziell das Verhalten als Herdenmitglied betreffen. Die Zugehörigkeit kann entweder statisch bei Programmstart vorgegeben sein oder sich als Resultat von Wahrnehmung und Verhaltensregeln ergeben.

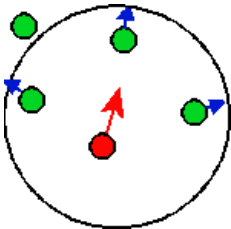
9.2.1 Ein Beispiel für die Regelmenge als Herdenmitglied



Strebe zur Mitte, falls die Herde zu weit weg ist



Halte Abstand zu den anderen Mitgliedern



Folge der allgemeinen Richtung

Diese Regeln sind anhand der Entfernung zur Herde verschieden gewichtet und erweitern die normalen Verhaltensregeln zur Wegfindung.

Kapitel 10

Gruppendynamik

Neben der Realisierung durch spezielle Regeln kann man zusätzlich eine Gruppe als einen einzigen komplexen Agenten auffassen, dem die einzelnen Gruppenmitglieder folgen. Ein Teil der Verhaltens wird dann nicht mehr von den verschiedenen Gruppenmitgliedern, sondern von der Gruppe als einzigem Stellvertreteragenten übernommen. Neben dem höheren Realismus durch einheitlicheres Verhalten sorgt dies gleichzeitig für eine Einsparung von Rechenzeit, insbesondere bei der Wegfindung.

Falls gewünscht, kann ein einzelnes Gruppenmitglied die Rolle des Anführers übernehmen. Dessen Auswahl kann auch durch Verhaltensregeln zur Laufzeit erfolgen. Dies ist sinnvoll, falls der Anführer verloren gehen kann oder andere Gruppenmitglieder sich im Laufe der Zeit durch veränderte Eigenschaften besser dafür eignen.

Die Verwendung expliziter Gruppen ermöglicht auch einfachere Initialisierungen mit generischen Regeln wie z.B. *60% Kinder, 20% Väter, 20% Mütter*. Für die Realisierung komplexer Hierarchien kann man Gruppen wiederum als Einheiten einer übergeordneten Gruppe verwenden.

Kapitel 11

Kontrolle der Simulation

Aufgrund der enormen Menge der Zustandsdaten aller Objekte ist es insbesondere zur Laufzeit kaum praktikabel, die Simulation mittels direkter Eingriffe in den Zustand einzelner Objekte zu steuern. Dazu kommt noch, daß die Ergebnisse solcher Manipulationen aufgrund von Interaktionen mit anderen Objekten letztendlich von den Erwartungen abweichen können. Andererseits gibt es im Alltag leistungsfähige sprachliche Begriffe zur Beschreibung des Verhaltens Einzelner und von Gruppen.

Es liegt somit nahe, Skriptsprachen um Befehle zu erweitern, die sich an den umgangssprachlichen Beschreibungen orientieren. In Verbindung mit selbst definierten Makros läßt sich dadurch der Aufwand zur Erstellung und Modifizierung eines Simulationslaufes auf wenige, leicht handhabbare Befehle konzentrieren.

Kapitel 12

Darstellung

Für experimentelle Zwecke genügt meist eine einfache schematische Visualisierung. Viele Echtzeitsimulationen, vor allem im Unterhaltungssektor, verlangen jedoch nach einer realistisch aussehenden Grafikdarstellung. Hier sind Methoden gefragt, die ohne allzu starke Qualitätseinbußen den Rechenaufwand der Darstellung begrenzen.

Da sich meist nur sehr wenige Agenten in unmittelbarer Nähe befinden, läßt sich die Polygonzahl gerendeter 3D-Objekte mittels LevelOfDetail-Techniken stark verringern. Indem man weiter im Hintergrund liegende Objekte gleicher Orientierung nur einmal in eine Textur rendert und per Billboarding mehrfach verwendet läßt sich der Aufwand weiter drücken. Hierbei ist allerdings auf eine genügend große Anzahl unterschiedlicher verwendeter Objekte zu achten, damit der Kunstgriff nicht störend auffällt.

Sowohl für die physikalische Modelle der Agenten als auch für die beim Billboarding verwendeten Texturen kann man vorberechnete Animationen einsetzen. Deren Anzahl ist ebenfalls ein Kompromiß aus Aufwand und erreichtem Realismus.

Für die graphische Ausgabe sind zwischen 30 und 70 Bilder pro Sekunde wünschenswert. Dagegen begnügt sich die dahinterstehende Simulation meist schon mit einer wesentlich geringeren Aktualisierungsrate. Dieser Unterschied läßt sich nutzen, indem man zwischen mehreren Simulationsschritten interpoliert. Auf diese Weise läßt sich sogar aus den diskreten Schritten eines zellulären Automats eine flüssige Animation zu erzeugen.

Kapitel 13

Adressen

- Sample Animations and Demonstrations:
www.research.rutgers.edu/~eaaron/demospage.html
- Zelluläre Automaten zur Simulation von Leben:
<http://members.cox.net/mathmistakes/simulating.htm>
- Mireks Celebration:
www.mirwoj.opus.chelm.pl/ca/index.html
- Crowd Control Simulation in Java Based Environment:
http://st.itim.unige.it/itim/websim1999_museum/w424.html
- Method of crowd simulation by using multiagent on cellular automata:
<http://csdl.computer.org/comp/proceedings/iat/2003/1931/00/19310046abs.htm>
- Interaction with Groups of Autonomous Characters:
www.red.com/cwr/
- Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model):
www.red3d.com/cwr/boids
- Hierarchical Model for Real Time Simulation of Virtual Human Crowds:
<http://csdl.computer.org/comp/trans/tg/2001/02/v0152abs.htm>