

Examensarbete



SVENSK TITEL

Undertitel

ENGLISH TITLE

Subtitle

Examensarbete inom huvudområdet Datavetenskap
Grundnivå 30 högskolepoäng
Vårtermin 2015

Arvid Backman

Handledare: Mikael Johannesson
Examinator: Anders Dahlbom

Sammanfattning

[Direkt efter titelsidan ska det finnas en sammanfattning, som omfattar högst 150 ord. Denna sida ingår inte i ordinarie sidnumrering. Sammanfattningen ska i kort form sammanfatta hela arbetet inklusive bakgrund, problemformulering, genomförande/implementation, resultat och framtida arbete. Den ska därmed vara det sista som skrivs i rapporten. Den behöver därför endast vara med i den rapport som lämnas in för opponering (och den slutgiltiga rapporten). I tidigare inlämningar kan denna text lämnas oförändrad.]

Nyckelord: [minst tre nyckelord men inte allt för många, helst inte fler än sex stycken]

Innehållsförteckning

1	Introduktion.....	1
2	Bakgrund.....	2
2.1	Artificiell intelligens.....	2
2.1.1	Traditionell AI	2
2.1.2	Spel-AI.....	2
2.1.3	Autonom agent	2
2.2	Styrbeteende	3
2.2.1	Sök	3
2.2.2	Ankomst	3
2.2.3	Väggundvikande.....	4
2.2.4	Vägföljning.....	4
2.2.5	Flödesfält.....	4
2.2.6	Flockbeteende.....	4
2.2.7	Separation	4
2.2.8	Sammanhållning.....	4
2.2.9	Formering	5
2.3	Beräkningsmodell	5
2.3.1	Viktad trunkerad summa	5
2.3.2	Viktad trunkerad summa med prioritering	5
2.4	Vägplanering.....	5
2.4.1	A*	5
3	Problemformulering	6
3.1	Problembeskrivning	6
3.1.1	Delmål 1: Implementation	6
3.1.2	Delmål 2: Utvärdering	7
3.2	Metodbeskrivning.....	7
3.2.1	Metod för delmål 1: Implementation.....	7
3.2.2	Metod för delmål 2: Utvärdering	7
3.3	Metodreflektion	7
4	Genomförande	8
4.1	Förstudie.....	8
4.2	Progressionsexempel: modellering	8
5	Utvärdering.....	9
5.1	Presentation av undersökning.....	9
5.2	Analys.....	9
5.3	Slutsatser.....	9
6	Avslutande diskussion.....	10
6.1	Sammanfattning.....	10
6.2	Diskussion	10
6.3	Framtida arbete	10
	Referenser	11

1 Introduktion

[Introduktionen ska innehålla en sammanfattning av bakgrund, problem, metod och genomförande/implementation. Inför varje inlämning behöver bara de relevanta delarna av introduktionen lämnas in. Tänk på att lämna plats i introduktionen för stycket om genomförande vid första inlämningen. Introduktionen bör revideras vid varje inlämning eftersom arbetet förändras över tid.

För att separera innehållet i introduktionen ska den indelas i onummerade stycken med hjälp av radbrytningar (i likhet med dokumentmallen). Styckena ska inte vara kortare än 4-5 rader text och innehålla ett sammanhållet ämne - till exempel bakgrund om datorspel.

Introduktionen ska *inte* ha numrerade underkapitel eller figurer och vara ungefär en sida lång (inte under en halv sida och inte mycket mer än en sida).

Introduktionen ska ha sidnummer 1.]

2 Bakgrund

2.1 Artificiell intelligens

Artificiell intelligens är konsten att skapa maskiner som utför uppgifter som kräver intelligens när de utförs av människor. (Kurzweil, 1990). Artificiell intelligens handlar om att få datorer att utföra uppgifter och handlingar som människor och djur är kapabla att göra (Millington & Funge, 2009). Det är möjligt att programmera en dator att utföra uppgifter som, i stort sett, är omöjliga för en människa att lösa. Såsom; sökning, aritmetiska problem, med mera. **DEFINITION AV AI**

Det är dock ett flertal uppgifter som datorer är dåliga på att utföra, som människor finner triviala; bestämma vad som ska göras härnäst, känna igen ansikten, vara kreativa är endast några få exempel. Det är just detta som AI-området utforskar genom att undersöka vilka algoritmer det är som krävs för att få fram dessa egenskaper hos datorer. **PROBLEM MED AI**

2.1.1 Traditionell AI

Det traditionella AI-området är uppdelat i två, mindre, områden: stark AI och svag AI. Stark AI eftersträvar att skapa ett beteende som efterliknar människors tankeprocess, medan forskningen inom svag AI applicerar AI teknologier på resultatet av verkliga problem. Buckland menar, i boken *Programming Game AI by Example* (2005), att dessa två subområden tenderar att fokusera på att lösa ett problem på ett optimalt sätt med mindre hänsyn till hårdvara eller tidsbegränsningar. Buckland fortsätter med att beskriva ett exempel och menar på att en AI-forskare kan låta simulation exekveras i timmar, dagar, eller veckor så länge det ger ett lyckat resultat som kan diskuteras i en artikel. **Definition av traditionell AI**

2.1.2 Spel-AI

Artificiell intelligens har alltid funnits inom datorspel, men Pacman (**REF PACMAN**) var det första spelet med en, relativt, avancerad AI. Fienderna rörde sig precis som spelaren och gav känslan av att det var riktiga människor som styrde dem (Millington & Funge, 2009). **Kort historia om spel-AI**

En väldigt stor procent av de spel som finns idag har någon sorts AI implementerad. Om det så är en hund som rör sig mellan två olika rum i ett hus eller om det är en mer avancerad NPC (Non-Player Character) i ett rollspel som rör sig runt i en by så har båda agenterna ett sätt att navigera sig genom den miljön de befinner sig i. Inom datorspel är navigering och rörelse av agenter ett vanligt problem, oftast inom spel där en grupp av agenter ska navigeras tätt intill varandra. Tätt navigerande agenter är väldigt vanligt inom RTS (realtidspel) så som (**STARCRAFT2**) och (**WARCRAFT3**). **Navigering i datorspel**

2.1.3 Autonom agent

En agent är någonting som kan uppfatta miljön runtomkring sig och agera utifrån den informationen. En agents beteenden beskrivs av agentfunktioner som mappar en given uppfattning till en mekanism (Russell & Norvig, 2010). **Vad är en agent?**

En autonom agent i ett datorspel är en enhet i en spelvärld som, till viss del, har egenskapen att improvisera sina beslut (C. Reynolds, 1999). Buckland definierar en autonom agent på följande sätt: **Definition av en autonom agent**

An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

Buckland, 2005, s.85

Reynolds beskriver i *Steering Behaviors For Autonomous Characters* (1999) att ett beteende hos en autonom agent kan delas upp i flera lager för att lättare förstå det. Dessa lager är: handlingsval, styrning, och förflyttning (**FIGUR**). **Autonoma agents beteende**

- Handlingsval: Lagret som bestämmer vilket mål agenten har. Till exempel: "gå hit".
- Styrning: Lagret som ansvarar för att navigera agenten korrekt för att tillgodose målen som gavs från det tidigare lagret. Detta uppfylls genom att applicera styrbeteenden hos agenten för att producera en styrkraft som beskriver hur agenten ska röra sig.
- Förflyttning: Det lager som ansvarar för en agents förflyttning. Detta lager konverterar kontrollsignaler från styrningslagret till rörelse av agentens kropp.

Ett exempel på en autonom agents beteende, i ett realtidstrategispel, är att spelaren ger order åt en autonom agent att röra sig till en ny position och den autonoma agenten navigerar sig själv genom miljön. Ett annat exempel är att ha två autonoma agenter, en råtta och en katt. Katten rör sig runt i en miljö medan musen sitter och äter. När musen ser att katten närmar sig flyr den från katten, samtidigt som katten börjar jag musen. Alla dessa beslut görs utan någon översyn av en programmerare eller spelare. **Exempel på en autonom agents beteende**

2.2 Styrbeteende

Ett beteende som appliceras för att producera en styrkraft hos en agent, kallas för ett styrbeteende. Det finns en mängd olika styrbeteenden som producerar en styrkraft på olika sätt. Flera av styrbeteendena kan kombineras för att styra den autonoma agenten på ett mer komplext och naturligt sätt. Dessa styrbeteenden presenteras av Reynolds i hans artikel: *Steering Behaviors For Autonomous Characters* (1999). **Vad är ett styrbeteende?**

2.2.1 Sök

Sök är ett styrbeteende som används för att styra en agent mot en specificerad position (P_{goal}), i global rymd. Beräkningen för detta styrbeteende är relativt enkel, först beräknas en önskad hastighet ($V_{desired}$) genom **FORMEL**. Med den önskade hastigheten går det att beräkna vilken styrkraft (F_{steer}) som ska appliceras på agenten genom **FORMEL**.

2.2.2 Ankomst

Styrbeteendet ankomst är identiskt till sök, så länge agenten är långt ifrån den specificerade målpositionen. Skillnaden från sökbeteenden, som styr sig mot målet, är att beteendet får karaktären att sakta ner när den närmare sig målet. Sträckan som definierar när agenten ska sakta ner går att förändra.

2.2.3 Väggrundvikande

Väggrundvikande ger en agent förmågan att styra ifrån potentiella kollisioner med väggar i en miljö. En vägg är ett linjesegment, i 3D en polygon. Detta görs med hjälp av avkännarpunkter för att se om en avkännarpunkt korsar en vägg. Om en avkännare korsar en vägg beräknas en styrkraft genom att beräkna hur mycket avkännaren har penetrerat väggen och sedan skapa en kraft av den penetrerade längden i väggens normalriktning.

2.2.4 Vägföljning

Vägföljningsbeteendet gör det möjligt för en agent att styras längs en serie av positioner som formar en väg. Agenten tittar, med hjälp av sin nuvarande position och hastighet, vilken position han kommer att ha om x sekunder (P_{future}). Från P_{future} beräknas den närmsta punkten på vägen (P_{path}). Ett sökbeteende appliceras sedan på agenten, med P_{path} som målposition. Om P_{path} skulle vara den sista punkten på vägen byts sökbeteendet ut mot ankomstbeteendet.

2.2.5 Flödesfält

Styrbeteendet flödesfält är ett beteende för navigering av agenter, och kan användas som ett alternativ till vägföljningsbeteendet. Ett flödesfält är ett, godtyckligt stort, rutnät, där varje cell i rutnätet innehåller en riktningsvektor. Denna riktningsvektor representerar vilken styrkraft som ska appliceras på en agent när den befinner sig i cellen. Cellernas riktningsvektorer kan vara statiska, men de kan också uppdateras dynamiskt. Dynamisk uppdatering av flödesfältet är fördelaktigt i spel där hinder för agenter förändras i realtid. Exempelvis realtidstrategispel, där agenterna inte får styras in i varandra och deras position alltid förändras.

2.2.6 Flockbeteende

I *Flocks, Herds, and Schools: A Distributed Behavioral Model* (1987) beskriver Reynolds tre olika styrbeteenden som alla samarbetar för att skapa ett flockbeteende hos grupper av agenter. Dessa tre styrbeteenden är: Separation, sammanhållning, och formering. Styrbeteendena appliceras endast på en agent beroende på de agenter som befinner sig i ett definierat närområde.

2.2.7 Separation

Separationsbeteendet skapar en kraft som styr en agent ifrån andra agenter inom sitt närområde. När detta beteende appliceras på en mängd agenter kommer dom sprida ut sig, och försöka maximera längden från varandra. Kraften (F_{steer}) beräknas genom att beräkna en riktningsvektor mot alla agenter i närområdet. Riktningsvektorerna normaliseras och adderas sedan, (**SE FORMEL!**). Detta beteende kan användas för att hindra agenter att tränga ihop sig.

2.2.8 Sammanhållning

Sammanhållningsbeteendet ger en agent förmågan att närma, och gruppera, sig med andra agenter i närområdet. Styrkraften (F_{steer}) kan beräknas genom att beräkna medelpositionen hos de närliggande agenterna. Styrkraften kan sedan appliceras i riktningen från agenten och medelpositionen, (**SE FORMEL**).

2.2.9 Formering

Formeringsbeteendet ger en agent förmågan att röra sig i samma riktning och hastighet som agenter i närområdet. Kraften (F_{steer}) kan beräknas genom att först beräkna medelhastigheten hos alla närliggande agenter och sedan subtrahera, denna medelhastighet, med agentens egen hastighet, se (**FORMEL**).

2.3 Beräkningsmodell

2.3.1 Viktad trunkerad summa

2.3.2 Viktad trunkerad summa med prioritering

2.4 Vägplanering

Den vanligaste metoden för att få agenter att navigera sig igenom en miljö, i ett datorspel, är genom vägplanering. Den huvudsakliga uppgiften som vägplaneringen står för, är att hitta den kortaste vägen mellan två definierade punkter. Vägplaneringen tar även hänsyn till statiska objekt i miljön, såsom byggnader, berg osv., när den kortaste vägen mellan punkterna beräknas. A* (Hart, Nilsson, & Raphael, 1968) är en av de vanligaste algoritmerna för att beräkna den kortaste vägen. **Vad är vägplanering och vilken är den vanligaste algoritmen som används för att hitta den kortaste vägen?**

2.4.1 A*

A*, även kallad A-stjärna, är en sökalgoritm som först framställdes av Peter Hart, Nils Nilsson, och Betram Raphael (1968). Algoritmen är en förbättring av Dijkstras algoritm (Dijkstra, 1959) och det som främst skiljer de två algoritmerna ifrån varandra, är att A* använder sig av heuristik för att hitta den kortaste vägen hos en graf. Heuristik innebär att algoritmen gör en uppskattning på vad avståndet från den nuvarande positionen till målpositionen. Funktionen för, att beräkna, det heuristiska värdet varierar från problem till problem och alla medför för- och nackdelar. **Historien hos A***

Anledningen till att A* är en vanlig algoritm för vägplanering inom dataspel är på grund av två egenskaper den har: Den första egenskapen är att algoritmen är komplett, vilket innebär att om det finns en väg till målet kommer den vägen att hittas. Den andra egenskapen är att algoritmen alltid kommer att hitta den mest optimala vägen, om det heuristiska värdet inte överskrider den verkliga kostnaden (Hart et al., 1968). **Varför är A* vanligt inom spel?**

3 Problemformulering

Denna del av examensarbetet är uppdelat i två rubriker: Problembeskrivning och Metodbeskrivning. Problembeskrivning redovisar det problem som examensarbetet är baserat på. Metodbeskrivning redovisar hur examensarbetets frågeställning ska besvaras, undersökas, och utvärderas.

3.1 Problembeskrivning

Detta arbete har som syfte att jämföra olika tekniker för navigering hos grupper av autonoma agenter inom realtidstrategispel. Examensarbetet kommer att fokusera på två olika tekniker för navigeringen: Flödesfältsnavigering och vägföljningsnavigering. Båda teknikerna kommer att slås ihop med andra styrbeteenden när de utvärderas. Detta kommer att ske med hjälp av en beräkningsmodell. Detta är främst applicerbart hos spel som går inom genren realtidstrategispel, såsom **(STARCRAFT)** och **(TOTAL ANNIHILATION)** men kan även användas inom andra spelgenrer, som till exempel rollspel där icke spelarstyrda karaktärer ska navigera sig i en miljö. **Vad är syftet med arbetet?**

Det kommer att utföras experiment på grupper av autonoma agenter och storleken på dessa grupper kommer att variera från ett litet antal till ett hundratal agenter mellan de olika testfallen. Eftersom att miljöer i spel kan variera kommer testfallen att köras på ett flertal olika miljöer som lägger fokus på vissa aspekter, som till exempel; trånga och fria utrymmen, korridorer, och dynamiska miljöer. Den frågeställning som arbetet kommer försöka besvara är: **Hur kommer testerna se ut, och vad ska dom besvara för frågeställning?**

- Hur förhåller sig flödesfältnavigering mot vägföljningsnavigering hos samlingar av autonoma agenter med avseende på effektivitet?

För att det ska vara möjligt att besvara frågeställningen kommer en applikation skapas. Det ska vara möjligt att förändra vissa värden i applikation, som till exempel rutnätets densitet för navigering med flödesfält. Det ska även vara möjligt att välja en miljö som testerna ska utföras på. **Hur ska applikationen se ut?**

I takt med att datorspel blir större och mer avancerade kommer antalet agenter i datorspel och komplexiteten på miljöer att öka. Detta kommer att ha en påverkan på hur effektiv navigeringen av agenterna är, men också hur estetiskt tilldragande de rör sig. Det är därför intressant att utföra tester för att se vilka navigeringstekniker som fungerar bäst under olika förhållanden. **Varför är frågeställningen intressant att besvara?**

Arbetets genomförande är indelat i två delsteg. Det första steget är att implementera en applikation vars syfte är att testa de två teknikerna, som arbetet ska utvärdera, i ett antal miljöer. Det andra steget är att göra utvärderingar på teknikerna med hjälp av den applikation som tidigare implementerats. **Förklara genomförandet av arbetet**

3.1.1 Delmål 1: Implementation

Avsikten med detta delmål är att implementera den applikation som kommer att användas för att utvärdera navigationsteknikerna. Därmed behöver alla styrbeteenden och beräkningsmodeller som ska användas, för utvärderingen, implementeras i applikationen. Det ska vara möjligt att skapa olika utfall för att enkelt kunna analysera teknikerna. Med olika utfall menas vilken miljö som ska användas, hur många agenter som ska navigera

genom vald miljö, och vilken teknik de ska använda. **Vad kommer behöva implementeras för att kunna besvara frågan?**

3.1.2 Delmål 2: Utvärdering

Detta delmål har som syfte att utvärdera navigeringsteknikerna. Det är i detta steg testfallen skapas med hjälp av applikation som implementerats i det tidigare delmålet. Vad som skiljer testfallen ifrån varandra är främst hur dess miljö ser ut, men även antalet agenter som navigerar genom miljön. **Skapandet av testfall**

Den egenskap som arbetet kommer att utvärdera hos de två teknikerna är deras effektivitet. Den operationella definitionen av effektivitet i detta arbete är tidseffektiviteten och navigeringstiden hos teknikerna. Med tidseffektivitet menas antalet sekunder det tar för tekniken att hitta en väg för agenterna. Navigeringstiden är en mätning på hur lång tid det tar för agenterna att nå sin måldestination. **Vad kommer att utvärderas**

3.2 Metodbeskrivning

3.2.1 Metod för delmål 1: Implementation

Metoden för detta delmål är att implementera en applikation vars uppgift är att kunna besvara den frågeställning som arbetet har. Den mest vitala delen i detta delmål är implementationen av flödesfält- och vägföljningsnavigering. Styrbeteendena, och den beräkningsmodell, som agenterna kräver för att navigera sig i miljön måste också implementeras. Det krävs även att applikationen har en implementation av sökalgoritmen A^* , för att kunna hitta en väg som vägföljningsbeteendet kan använda. **Hur ska implementationssteget genomföras**

3.2.2 Metod för delmål 2: Utvärdering

Metoden för denna del är att utvärdera de testfall som skapats i applikationen från tidigare del. För att göra det möjligt att utvärdera egenskapen av navigering kommer det att används två mått. Navigeringen går ut på att en grupp agenter ska ta sig från en punkt till en annan i en arbiträr miljö. Dessa två mått är: **Vilka mått ska användas för att utvärdera teknikerna**

- Hur många sekunder tar det att generera en väg för agenterna?
- Hur många sekunder tar det för agenterna att ta sig till målpunkten?

För att utvärdera dessa mått hos teknikerna kommer ett flertal testfall utföras på de två teknikerna, tillsammans med styrbeteendena och en beräkningsmodell. För att utvärderingen ska bli så tydlig som möjligt kommer måtten hos teknikerna att jämföras och analyseras för att få fram vilken av teknikerna som skulle kunna lämpa sig bäst för navigering hos grupper av agenter. **Hur ska måtten utvärderas**

3.3 Metodreflektion

4 Genomförande

4.1 Förstudie

4.2 Progressionsexempel: modellering

5 Utvärdering

5.1 Presentation av undersökning

5.2 Analys

5.3 Slutsatser

6 Avslutande diskussion

6.1 Sammanfattning

6.2 Diskussion

6.3 Framtida arbete

Referenser

- Buckland, M. (2005). *Programming Game AI by Example* (p. 495).
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. doi:10.1007/BF01386390
- Hart, P., Nilsson, N., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107. doi:10.1109/TSSC.1968.300136
- Kurzweil, R. (1990). The age of intelligent machines. Retrieved from <http://dl.acm.org/citation.cfm?id=85231>
- Millington, I., & Funge, J. (2009). *ARTIFICIAL INTELLIGENCE Second Edition* (p. 895). doi:10.1017/S0263574700004070
- Reynolds, C. (1999). Steering Behaviors For Autonomous Characters. Retrieved from <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.8035>
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87* (Vol. 21, pp. 25–34). New York, New York, USA: ACM Press. doi:10.1145/37401.37406
- Russell, S. J., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach. Retrieved from http://books.google.se/books/about/Artificial_Intelligence.html?id=8jZBksh-bUMC&pgis=1

Appendix A - Designdokument etc.

[Appendix ska fungera som referenslistan - dvs det ska finnas referenser till den från texten. Appendix ska inte vara numrerade utan ska namnges med: Appendix A, Appendix B osv. De ska vara sidnumrerade (I, II, III ...) men de ska inte finnas med i innehållsförteckningen. Varje nytt appendix ska börja på toppen av sidan.]