Linköping Studies in Science and Technology

Dissertations No. 1477

Affective Decision Making in Artificial Intelligence

Making Virtual Characters With High Believability

Anja Johansson

Linköping University
INSTITUTE OF TECHNOLOGY

Department of Science and Technology
Linköping University

Norrköping 2012

Affective Decision Making in Artificial Intelligence:
Making Virtual Characters With High Believability

Anja Johansson

Cover Image:

Networks in a nutshell
by Anja Johansson

Copyright © 2012 Anja Johansson, unless otherwise noted.

Figures 4.1, 4.2, 4.11, 4.12, 4.13, 4.14, 4.15 & 4.16 are reprinted with kind permission from Springer Science and Business Media.

Printed by LiU-Tryck, Linköping 2012

ISBN 978-91-7519-792-0 ISSN 0345-7524

Abstract

Artificial intelligence is often used when creating believable virtual characters in games or in other types of virtual environments. The intelligent behavior these characters show to the player is often flawed, leading to a worse gameplay experience. In particular, there is often little or no emotional impact on the decision making of the characters.

This thesis focuses on extending decision-making and pathfinding mechanisms for virtual characters, with a particular focus on the use of emotions. The thesis is divided into three parts.

The first part is an introductory study concerning the requirements designing a believable virtual character places on the architecture used. Gameplay design patterns are used as a tool to analyze the proposed agent architecture and discussions are presented regarding the necessary properties of such an architecture with respect to gameplay.

The second part extends two action selection mechanisms to include emotional impact. In particular, behavior networks are extended to take complex emotional impact into account, including emotional parameters, emotional goals, and emotional influences. Moreover, time-discounting is introduced into behavior networks as a factor in the decision making. The time-discounting is also under emotional influence. The second action selection mechanism extended to use emotional impact is behavior trees. Since behavior trees are widely used by game designers, allowing full control over the characters' behaviors, the work in this thesis proposes a new type of emotional selector which only affects a part of the behavior tree, leaving the control in the hands of the designer.

The third part focuses on more complex pathfinding where more factors than finding the shortest collision-free path through an environment are considered. A new type of visibility map is introduced. Using the knowledge of the virtual character about previous enemy positions, a more accurate visibility map is created. The visibility map is used for covert pathfinding, where the character tries to find a path through an environment while trying to minimize the risk of being seen by the enemy. Finally, a new kind of pathfinding, emotional pathfinding, is introduced, based on the use of emotion maps. Humans often have emotional attachment to geographical locations because they have previously felt emotions at those locations. This approach takes advantage of this knowledge and enables a virtual character to find a path through an environment that is as emotionally pleasant as possible.

Populärvetenskaplig Sammanfattning

Artificiell intelligens används ofta för att skapa virtuella karaktärer för spel eller andra typer av interaktiva installationer. Dessa karaktärer borde visa ett intelligent beteende, men ofta är deras beteende felaktigt vilket kan leda till en sämre spelupplevelse.

Denna avhandling fokuserar på att utöka karaktärernas beslutsprocesser och stigfinnandesystem, med ett speciellt fokus på användningen av känslor.

I den första delen av avhandlingen beskrivs en inledande studie vars mål var att analysera karaktärernas trovärighet och hur denna trovärdighet påverkas av vilken typ av arkitektur man använder. Designmönster för spel används för att analysera en viss arkitektur. Därefter diskuteras vilka egenskaper hos arkitekturen som är nödvändiga för att uppfylla dessa designmönster.

I den andra delen utökas tvåbeslutsmetoder genom att introducera känslor. Beteendenätverk utökas så att de inkluderar bl.a. känslosamma parametrar, känslosamma mål, och känslosamma influenser. Dessutom introduceras tidsuppfattning tillsammans med en känslokomponent i beteendenätverken. Den andra beslutsmekanismen som utökas med en känslokomponent är beteendeträd. Beteendeträd används till en stor utsträckning av speldesigners eftersom de tillåter full kontroll över karaktärernas beteenden. Därför föreslås det i denna avhandling en ny typ av känslonod som innebär att kontrollen förblir hos speldesignern.

Den tredje och sista delen fokuserar på att skapa en mer komplex typ av stigfinnandesystem. Många algoritmer fokuserar enbart på att hitta den kortaste vägen till målet. Algoritmerna i denna avhandling använder sig i stället av synlighet och känslor för att styra stigfinnandet. Synlighetsalgoritmen baseras på karaktärerns tidigare kunskap om var fiender befinner sig. En synlighetskarta skapas utifrån denna kunskap och kartan används sedan i stigfinnandet för att hitta en väg där karaktären kan undgå att bli sedd. Känsloalgoritmen skapar känslokartor baserat på vilka känslor karaktären upplevt i de olika områdena. Detta är baserat på vetskapen att människor ofta knyter an känslor till geografiska positioner beroende på vilka känslor de haft där. Känsloalgoritmen använder känslokartorna i stigfinnandeprocessen för att hitta en stig som är så angenäm som möjligt.

Acknowledgements

I wish to give my warmest thanks to my supervisor, Pierangelo Dell'Acqua, for his support, guidance and encouragement during these five years. Without our discussions and brainstorming meetings this thesis work would not have been possible.

Another person whom I wish to thank is my co-supervisor, Stefan Gustavson. He has been instrumental in increasing my motivation through his supervision and inspirational discussions.

I would like to show my gratitude towards Eva Skärblom and Gun-Britt Löfgren for their patient help with the numerous conference bookings and travel arrangements throughout these years. Making bookings for someone with travel anxiety is not always an easy chore but you have succeeded wonderfully.

I also wish to thank Katerina Vrotsou, Aida Nordman, and Ivan Rankin for their invaluable help with comments on this thesis. I swear I will learn to write properly one day.

A big round of thanks goes to my friends, both near and far, for having my back during these years. You soon figured out that the question "How is the thesis work going?" is taboo. You kept me sane during the last 6 months of frantic preparations and I am eternally grateful!

Finally, I would like to give a big bunch of kisses to the members of my family, who never stopped believing in me. You are the best. Thank you so much!

List of Publications

The following papers are included in the thesis:

i Anja Johansson and Pierangelo Dell'Acqua. *Affective States in Behavior Networks.* In Dimitri Plemenos and Georgios Miaoulis, editors, Intelligent Computer Graphics 2009, volume 240 of Studies in Computational Intelligence, chapter 2, pages 19–39. Springer Berlin / Heidelberg, 2009.

ii Anja Johansson and Pierangelo Dell'Acqua. *Introducing Time in Emotional Behavior Networks.* In proceedings of 2010 IEEE Conference on Computational Intelligence and Games, CIG'10, pages 297–304, Copenhagen, Denmark, August 18–21 2010.

iii Anja Johansson and Pierangelo Dell'Acqua. *Knowledge-Based Probability Maps for Covert Pathfinding.* In Ronan Boulic, Yiorgos Chrysanthou, and Taku Komura, editors, Motion in Games, volume 6459 of Lecture Notes in Computer Science, pages 339–350. Springer Berlin / Heidelberg, 2010.

iv Petri Lankoski, Anja Johansson, Benny Karlsson, Staffan Björk, and Pierangelo Dell'Acqua. *AI Design for Believable Characters via Gameplay Design Patterns.* Business, Technological, and Social Dimensions of Computer Games: Multidiciplinary Developments, chapter 2, pages 15–31. IGI Global, 2011.

v Anja Johansson and Pierangelo Dell'Acqua. *Pathfinding with Emotion Maps.* In Dimitri Plemenos and Georgios Miaoulis, editors, Intelligent Computer Graphics 2011, volume 374 of Studies in Computational Intelligence, pages 139–155. Springer Berlin / Heidelberg, 2012.

vi Anja Johansson and Pierangelo Dell'Acqua. *Comparing Behavior Trees and Emotional Behavior Networks for NPCs.* In proceedings of 17th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games, CGAMES'12, (to appear), 2012

vii Anja Johansson and Pierangelo Dell'Acqua. *Emotional Behavior Trees.* In proceedings of IEEE Conference on Computational Intelligence and Games, CIG'12, (to appear), 2012

The following papers are not included in the thesis but are related to the work:

viii Anja Johansson and Pierangelo Dell'Acqua. *Realistic Virtual Characters in Treatments for Psychological Disorders - an extensive agent architecture.* In SIGRAD'07: Computer Graphics in Healthcare, pages 46–52. Linköping University Electronic Press, November 2007.

ix Anja Johansson and Pierangelo Dell'Acqua. *Affective States in Behavior Networks.* In proceedings of 12th International Conference on Computer Graphics and Artificial Intelligence, 3ia'09, pages 19–32, Athens, Greece, May 29-30 2009.

x Anja Johansson and Pierangelo Dell'Acqua. *Pathfinding with Emotion Maps.* In proceedings of 14th International Conference on Computer Graphics and Artificial Intelligence, 3ia'11, pages 85–96, Athens, Greece, May 27-28 2011.

Contents

Introduction 1

Artificial intelligence (AI) is a branch of computer science that aims to create "intelligent" machines. It is a vast field, dealing with a variety of problems, such as reasoning, planning, learning, and knowledge representation. One of the major fields of AI involves creating intelligent, human-like virtual characters situated in virtual worlds, including computer games. This is the field this thesis makes contributions to.

1.1 Motivation

Over the past few decades computer games have become increasingly popular and now have budgets that reach tens of millions of dollars and more. The rapid development of computer graphics has greatly influenced the gaming industry, creating more realistic-looking and stunning game worlds. However, while the graphics have greatly improved over the years, the AI techniques that control the non-player characters have not undergone a similar drastic improvement. The difference between the realistic graphics and the rather flawed AI is often all too evident to players of computer games. This creates a discord that breaks the feeling of immersion and lessens the gameplay experience.

The motivation behind the research presented in this thesis is to be able to create more interesting and believable characters for virtual environments. A large part of these characters are non-player characters (NPCs) in games. They are characters that inhabit the game world and are not controlled by human players. Many games have a high level of graphical realism, but the behavior of the NPCs is often frustrating to the player. Artificial intelligence for virtual characters is developed in two rather distinct areas; commercial game companies and academia. Unfortunately, the two areas are not so cooperative and do not necessarily share the same goals, often leading to specific algorithms being used in industry and other algorithms in academia. The results from the academic research are available to those who have subscriptions to the journals/databases. Industry information is usually kept secret and only shared between companies. This thesis focuses on creating interesting behavior for virtual characters in games and interactive applications, but nevertheless has a foundation in academia. Therefore it is not always possible to make comparisons/integrations with algorithms currently used in industry.

The thesis concerns other types of virtual characters as well, not only NPCs. In general, the contributions of this thesis can be used wherever there is a need for believable characters with a rich personality. In games this is mostly useful in role-playing games, such as Oblivion or Skyrim. Other types of games, such as first-person shooters, tend to be more fast-paced and focus less on the individual NPCs. Other application areas, including serious games, will find the contributions of this thesis useful.

The work in this thesis can essentially be divided into three parts:

- The first part concerns believability in NPCs. It consists of using the concept of gameplay design patterns to define how well a proposed architecture fulfills the needs to enhance believability.
- The second part involves high-level emotional decision making. Two different types of decision-making mechanisms, behavior networks and behavior trees, are developed and expanded. Major emphasis is placed on the use of emotions to enhance the behavior and mimic the human decision-making process.
- The third part focuses on elaborate pathfinding that takes emotions and visibility into account, as well as the distance. Specifically, the contribution to pathfinding presented in this thesis proposes the use of emotion maps and visibility maps.

Believability, decision making, and pathfinding are three key concepts underlying the work presented in this thesis. Next, these concepts, the research challenges they pose, and the problems tackled by this thesis work, are presented.

1.2 Believability

The believability of a virtual character is difficult to define. Generally, what we mean by believability is the consistency of the behaviors of the character, i.e. that the decisions the character takes are understood by the viewer. Believability, the way it is defined in this thesis, concerns itself with how natural the behavior of a character is, how far the actions of the character agree with what the player believes should happen in such a context. Realism, on the other hand, has more to do with the appearance of the character; animations, 3D model, textures, etc. While realism is often high in modern commercial games, the believability is flawed at times. Realism can also refer to the scientifically accurate replication of human cognitive processes. In contrast, believability only requires that the behavior *appears* human-like[1]. The actual algorithms and methods used are irrelevant.

There are several challenges when it comes to believability. First of all, there is the challenge of how to specifically define believability in a computer game, and if/how it can be measured. Moreover, what requirements does believability pose on the choice and

[1] The behavior must only be human-like if the character is human. If the character is e.g. an alien, a dog, or a rabbit, naturally the behavior is expected to be very different in order to achieve believability.

development of an agent architecture? Can believability be accomplished by a simple reactive system or are more complex decision-making mechanisms necessary? Human behavior is often changing, intuitive, and difficult to replicate. A simple “normal” reaction to an event, a reaction that every human would understand, can be difficult for a computer to achieve. The challenge lies in replicating these very crucial behaviors that are such a big part of human behavior that they cannot be omitted or the believability is lost.

This thesis specifically addresses the question of analyzing what kind of architecture is needed to create believable virtual characters.

1.3 Decision Making

Decision making is one of the major areas of artificial intelligence. Making decisions is the key feature of any independent virtual character. A virtual character usually has a limited set of actions it can perform. Without a decision mechanism, there can be no reasonable behavior. There are many different decision-making algorithms for virtual characters, some focusing on planning, others on reactivity[2], and other again on a mixture of the two. For most scenarios where virtual characters are used, the decision-making algorithms must be fast enough to allow for real-time decision making. It is also important that the characters are able to react to the current events around them in a proper way.

A challenge here is the ability to merge reactivity and planning into one action selection mechanism. Virtual characters, and NPCs in particular, must be able to act without hesitation to events that occur. This requires reactivity. If an NPC is attacked it must automatically defend itself and not continue to do whatever it was doing prior to the attack. However, only reactivity gives very little believability, making the character’s behavior uninteresting and unintelligent. A major limitation of reactivity is that the character cannot think ahead, but simply waits for some stimuli from the environment that triggers a reaction. To be able to perform better and to give the player a sense of believability and immersion, the virtual character must be able to plan ahead, predict the results of its actions, and act as if it has its own set of goals to achieve. The combination of reactivity and planning is not trivial.

Another important problem is the sometimes over-rational behavior of virtual characters. The decision-making models used in industry and academia rarely fully incorporate emotions, feelings, and moods in a way that mimics human behavior. When emotions are used, they are often used as simple conditions used to switch between two behaviors. In contrast, theories in psychology claim that emotions affect our decision making in far more intricate and complex ways. Excluding emotions from decision making potentially makes it difficult to achieve the human-like behavior that is so often desired. A decision-making model where emotions are fully integrated is needed.

This thesis addresses the problem of fully incorporating emotions in decision-making mechanisms.

[2] In this thesis, reactivity is defined as a process that maps current stimuli to an action. In other words, neither previous knowledge nor future consequences are considered.

1.4 Pathfinding

Characters that walk right through buildings in a virtual environment are not very believable. Avoiding collisions with solid objects should be a primary goal. Pathfinding for games is usually concerned with finding a collision-free path through the environment from a starting point A to a target point B. There are, however, many aspects to a path that should be taken into consideration. Is the path natural-looking? Is this how a human being would walk if placed in a similar environment? In real-time simulations it is also crucial that the pathfinding is fast so that it can be updated often, if changes in the environment occur.

Finding the shortest collision-free path through an environment has been the focus of pathfinding research for a long time and numerous solutions have been presented to this problem. The challenge lies in creating natural-looking paths, paths that humans or animals could walk without observers thinking it odd-looking or strange. What are the things that affect us when walking through an environment and how do these things affect us? The reasons why a person chooses a certain route can be several; the scenic beauty, shelter from the wind, the openness of the landscape, interesting buildings, etc. All these should be incorporated into the pathfinding to create interesting and natural paths.

This thesis addresses the problem of creating a more sophisticated, human-like pathfinding, using emotion maps and visibility maps.

1.5 Terminology

This section contains a few clarifications regarding terminology that can be useful to the reader of this thesis.

Virtual Character A virtual character is an entity with animal-like or human-like properties, existing in a virtual world.

NPC NPCs, or non-player characters, are characters that exist in a game, but are not controlled by any of the humans playing the game. NPCs may also refer to non-player characters controlled by the gamemasters in traditional non-computerized role-playing games. These types of NPCs, however, are not addressed in this thesis. Furthermore, the term NPC is interpreted as encompassing all non-player characters, not merely, as is often the case among game players, non-player characters that are friendly or neutral to the human player.

Agent By the term agent this thesis refers to an *autonomous software 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." [FG97]. In this thesis this term most often implies a virtual character.

Emotional State By emotional state this thesis refers to how an entity is currently feeling, i.e. the sum of its emotions at a certain moment in time. It should not be confused with the term state in finite state machines or in behavior networks.

Affective Appraisal Affective appraisal refers to the process in which events from the environment are evaluated in terms of their emotional significance with respect to the agent's goals and priorities.

1.6 Overview of Included Papers

This section presents an overview of the papers included in this thesis work. A more thorough presentation of the papers can be found in Chapter 4. Unless stated otherwise, the author of this thesis is the first author of the papers and the main contributor.

Paper A The work in this paper introduces emotions into extended behavior networks using a psychological model by Loewenstein and Lerner [LWHW01] of how emotions affect decision making. The author of this thesis has done the modeling, implementation and testing of the proposed extension.

Paper B The work in this paper focuses on analyzing an agent architecture from the perspective of believability in NPCs. Gameplay design patterns are used as a tool to accomplish this. The author of this thesis is the second author of the paper and has contributed to the analysis of the architecture.

Paper C An extension to emotional behavior networks that also includes time is proposed in the paper. Furthermore, a new relevant concept of emotional time-discounting is introduced. The author of this thesis has done the design, completion, and testing of the time-extended emotional behavior networks.

Paper D The work in the paper proposes the use of knowledge-based visibility maps as a way to perform covert pathfinding. The author of this thesis has performed the design, implementation, and analysis of the knowledge-based visibility maps.

Paper E In this paper the concept of emotion maps as a tool to perform emotional pathfinding is presented. Emotion maps represent the character's previous emotional state in relation to the environment. The author of this thesis has carried out the modeling, implementation, and testing of the emotion maps.

Paper F The work in this paper compares emotional behavior networks to behavior trees in terms of functionality and management. Moreover, the paper discusses in which contexts to best use the two models. The paper is the result of a general discussion with the coauthor; the author of this thesis has carried out most of the comparison work.

Paper G The work in the paper proposes an extension of behavior trees to include emotional impact. Specifically, the paper suggests the use of three important factors in decision making, namely risk, time, and planning effort. The author of this thesis has created the model, the implementation, and carried out the experiments of the proposed extension to behavior trees.

1.7 Thesis Overview

The other parts of the thesis are the following. Firstly, a background chapter is presented for readers who wish to learn more about the techniques and theories that are the foundation of this work. Secondly, the agent architecture developed during the course of this thesis work is briefly introduced. Thirdly, the contributions of this thesis are presented in detail. Finally, there is a concluding discussion and future areas of research are discussed.

Background 2

This chapter contains background information that is relevant for understanding the thesis contributions. The chapter is divided into three parts: decision making, emotions, and pathfinding.

2.1 Decision Making for Virtual Characters

A large part of artificial intelligence is devoted to decision making. Our behavior is what defines us and behavior is no less important for robots, virtual characters or avatars. Since the focus of this thesis is on creating interesting and believable virtual characters for games or game-like scenarios, this section will begin by briefly describing two methods often used in games: finite state machines and hierarchical finite state machines. Next, behavior trees and behavior networks are described in great detail as they form part of the backbone of the thesis.

2.1.1 Finite State Machines

Finite state machines (FSMs) have been used extensively within the game industry to control NPCs [Mil06]. They are fast, simple to implement, and easy to understand.

A finite state machine consists of a set of states and a set of transitions. The states represent behaviors that the character can perform. The transitions represent conditions under which a character can move from one state to another. An example of an FSM can be seen in Figure 2.1. This FSM illustrates the decision-making for an NPC soldier. The soldier starts by patrolling the area. If the soldier sees an enemy, it will attack it. If during the attack the enemy dies, then the soldier goes back to patrolling. Instead if the soldier appears to be losing the fight then it escapes from the enemy. Once in a safe place, the soldier takes up patrolling the area once more.

The problem with finite state machines is that they do not scale up easily [Cha07a]. It gets difficult to manage the transitions between states as the number of states grow. If one is not careful with the design of the finite state machine, it is possible for the character to end up in a loop or a in single state that has no transitions leading out of it.

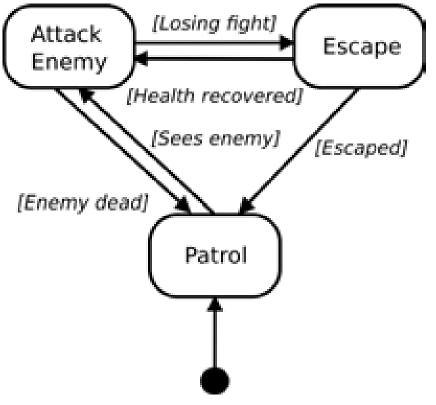


Figure 2.1: An example of a finite state machine.

2.1.2 Hierarchical FSMs

Hierarchical finite state machines (HFSMs) are similar to finite state machines, but allow grouping states together to form hierarchies. This makes it easier to share transitions between several states. An HFSM is depicted in Figure 2.2. While the soldier has enough health it will use the FSM to the right. This FSM is similar to the one mentioned in Section 2.1.1. The character will patrol the area, attack potential enemies and escape if necessary. However, should its health get too low at any point it will switch to the FSM to the left where it goes to a health well and drinks from it until its health has been restored. There is great benefit in using an HFSM here instead of designing the same behavior using an FSM, since in the latter case the link "health low" would need to be connected to all the states in the FSM to the right. When the size of the individual FSMs is large, the number of individual links can be greatly decreased using the hierarchical approach.

2.1.3 Behavior Trees

In the past few years, behavior trees have become an increasingly popular action selection mechanism for NPCs in commercial games. Games known to use behavior trees are Halo 2 [Isl05], Halo 3 [Isl08], Spore [Hec07], and GTA[Cha08b]. They were introduced into games to tackle the problems with FSMs and HFSMs, namely the problem with scaling up and with reusing behaviors.

Behavior trees do not seem to have a formal definition, possibly due to their use mainly within industry. While the basic algorithm remains roughly the same throughout the different implementations, the types of nodes tend to differ somewhat. Champandard [Cha07b, Cha07c, Cha08a, Cha08b, Cha09, CDHC10] has written extensively about behavior trees on his site AiGameDev.com. Knafla has also described behavior trees in detail [Kna11].

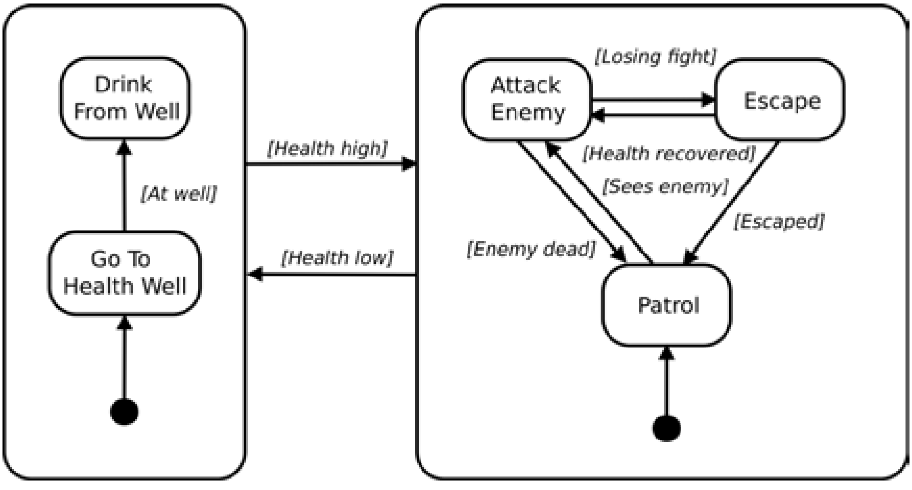


Figure 2.2: An example of a hierarchical finite state machine.

A behavior tree is a *directed acyclic graph* (DAG) consisting of different types of nodes. Most of the time, the graph is tree-shaped, hence the name behavior tree. In contrast to a usual tree, a node can have several parents, enabling the reuse of parts of the tree. The traversal of a behavior tree starts at the top node. When a node executes, it can return *success*, *failure* or *running*. While the first two are self-explanatory, the state *running* signifies that the node needs more time to execute. The most commonly used nodes in behavior trees, as described by Champandard and Knafla [Cha07b, Cha07c, Cha08a, Cha08b, Cha09, CDHC10, Kna11], are listed below.

Exterior/Leaf Nodes

Action An action represents a behavior that the character can perform, such as *shoot enemy* or *pick up food item*. An action that needs more time to complete returns the state *running*. In the figures of behavior trees in this thesis, actions are represented as white, rounded rectangles.

Condition A condition consults the knowledge of the character, returning *success* or *failure* depending on whether the condition holds. An example of a condition is *enemy close*, which implies checking if the distance to the closest enemy is lower than a certain threshold. In the figures, conditions are represented as gray, rounded rectangles.

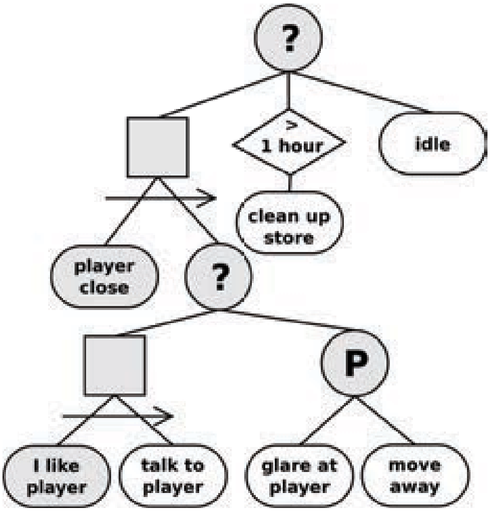


Figure 2.3: An example of a behavior tree.

Interior Nodes

Sequence Selector A sequence selector has a set of child nodes that it tries to execute in a given sequence. If one of the child nodes fails, the sequence selector returns *failure*. If the sequence selector has successfully executed all the child nodes, it returns *success*. If a child node returns *running*, the sequence selector will also return *running*, and it will remember which child node it should continue to execute in the next cycle.

In the figures, the sequence selector is represented by a grey square with an arrow across the lines connecting to its child nodes.

Priority Selector A priority selector tries to execute its child nodes, which are ordered according to a fixed priority, one at a time. The priority selector remembers which of its child nodes, if any, is currently *running*. If a child node succeeds, the priority selector terminates with a *success*. If none of the child nodes executes successfully, the priority selector returns *failure*.

In the figures, the priority selector is represented as a grey circle with a question mark in it. Furthermore, the priority is ordered from left to right.

Parallel Node A parallel node executes all of its child nodes in parallel. One may specify for each parallel node the number of child nodes required to succeed for the parallel node to succeed. Likewise, one may specify the number of child nodes that must fail in order for the parallel node to fail. In the figure, the parallel node is represented as a gray circle with a P in it.

Decorator A decorator is a node that acts as a kind of filter upon its single child node, placing extra constraints on the execution of its behavior without altering the original node. For instance, a decorator may prevent an action from being executed more often than once every five minutes.

In the figures, the decorator is depicted as a diamond with descriptive text.

An example of a behavior tree is depicted in Figure 2.3. The NPC for which this behavior tree is designed can be thought of as a friendly NPC situated in grocery store in a role-playing game. The traversal of the tree starts at the top. If the player is close by, the NPC will either talk to the player (if the NPC is fond of the player) or move away from the player while glaring at him/her at the same time. If the player isn't close and the store hasn't been cleaned in the last hour the NPC will clean the store. Otherwise, the NPC does nothing in particular.

2.1.4 Behavior Networks

In 1989, Maes suggested an energy-driven approach to decision making [Mae89] named MASM (Maes' Action Selection Mechanism). MASM is a distributed non-hierarchical network. The model was created to combine planning and reactive properties in one system. In her model, activation spreads from the goals of the agent as well as from the sensor input (often tied to the knowledge about the environment the agent is situated in).

Tyrrell has done an extensive evaluation [Tyr94] of MASM, describing several problems with the method. In MASM, there is a division scheme for activation, using the number of links leading to and from a module. This gives prejudice against nodes that receive energy from sensors which also affect other nodes.

In 1999, Dorer extended Maes' work and addressed the issues described by Tyrrell. Dorer extended behavior networks to continuous domains [Dor99a, Dor99b], and in 2004 he included resources to enable parallel behavior execution [Dor04]. These extensions of MASM go by the name of *extended behavior networks* (EBNs). EBNs have been used extensively in RoboCup (an international robot soccer competition) competitions [Dor99b] by the magma Freiburg team. Pinto and Alvares have also used EBNs as the action selection mechanism for NPCs in Unreal Tournament [PA05a, PA05b] with good results. Since a relevant part of this thesis is based on the work by Dorer, a thorough overview of EBNs will be given here. An EBN consists of a set of goals, a set of states[1], a set of resources, a set of behavior modules[2] and finally a set of parameters. In EBNs, each update cycle (cycles should occur frequently to give fast reactions to environmental changes) *activation* is spread from the goals of the network to behavior modules, and also from behavior modules

[1] Dorer does not specifically define states as a separate set, but the work in this thesis extracts them as a separate set for convenience. It is especially useful for the behavior network designer to view states separately when states are shared between behavior modules. Dorer uses the term propositions instead of states.

[2] Dorer names these competence modules.

to behavior modules[3]. The activation of a behavior module can be seen as the utility of the behavior. The more activation, the more desirable it would be to perform that behavior. However, behaviors cannot be chosen simply because they are desirable. It must also be possible to perform them. Preconditions of behavior modules are taken into account when deciding which behavior to perform.

The different nodes in the network and the activation spreading are described in detail below.

States

A state represents a belief the agent has about something in the environment (such as perceiving an enemy) or some internal state (such as the level of hunger). In behavior networks, states are represented as continuous values in the range $[0, 1]$.

In the figures, states are represented as rectangles.

Resources

A resource represents a necessary means to execute a behavior. Resources are often physical entities, such as body parts. Each resource is coupled with the number of available units and the number of bound units (units of this resource that are currently being used by behaviors).

In this thesis, resources are left out of the figures to avoid cluttering.

Goals

A goal is linked to one or more states that the agent wishes to accomplish, such as "enemy dead". Each goal g has one or more such conditions that need to be fulfilled in order for the goal to be achieved. A goal has a static importance, I_s. The static importance is a fixed value in the range $[0, 1]$. A goal can also have a dynamic importance, I_d, which is linked to a state. This enables a goal to have a varying importance depending on the value of the state. For instance, the goal "get rich" may be more important the less money the character has. The importance I_g of goal g is calculated as

$$I_g = f(I_s, I_d)$$

where f is any continuous triangular norm (e.g. multiplication) [KMP00].

In the figures, goals are depicted as octagons with the conditions attached to the bottom arc, and the dynamic importance attached to the left arc.

Behavior Modules

A behavior module represents an *action* that can be performed by the agent, such as "go to work". A behavior module can have one or more *preconditions* (represented by states).

[3]Note that Dorer does not let activation spread from stimuli to behavior modules, in contrast to MASM.

A behavior module also has a list of *effects*. Each one of the effects contains a possible outcome of the action coupled with the probability of that outcome. Note that the outcome can be a state or the inverse of a state. For example, if there is a state in the behavior network that is called “holding the ball”, the action “drop the ball” will lead the inverse of the state “holding the ball”.

In the figures, the behavior module is depicted as a circle, with preconditions attached to the lower arc, and effects attached to the top arc.

Activation Spreading

Every behavior module may receive activation from each goal in the network. The activation is spread from the goal to the behavior modules. In turn, the behavior modules spread activation internally, from module to module. The total activation for each behavior module is calculated and used to select which behavior to execute. The activation spreading in the behavior network is controlled by the following parameters:

- γ - the activation influence determines how much activation is spread through positive links[4]. Positive links are depicted as solid green arrows in the figures throughout this thesis. An example can be seen in Figure 2.4.
- δ - the inhibition influence determines how much activation is spread though negative links. Negative links are depicted as red dashed arrows.
- β - the inertia of the activation determines how much the activation during the previous activation spreading cycle affects the current activation.
- θ - the global activation threshold determines the initial threshold the execution-value must exceed for the behavior to be performed.
- $\Delta\theta$ - the threshold decay determines how much the threshold should be lowered between each cycle if no action can be selected.

The parameters γ, δ, and β must lie in the range $[0, 1[$. The parameter θ must lie in the range $[0, \hat{a}]$, where $\hat{a}$ is the maximum value the activation of a behavior module can reach. The parameter $\Delta\theta$ lies in the interval $]0, \hat{a}[$[5].

Each activation cycle t, activation propagates from the goals to the behavior modules. There are four ways by which a behavior module can receive activation.

1. A behavior module k receives positive activation $a^t_{kg'}$ from a goal g with importance I_g if one of the effects of k is one of the conditions of g:

[4]Positive links occur when the effect of a behavior module is the same as the precondition of another behavior module. Likewise, negative links signify a link where the effect of a behavior module is the opposite of the precondition of another behavior module.

[5]Some implementations of behavior networks use percentages, but Dorer uses a fixed value.

$$a_{kg}^{t}\prime = \gamma \cdot I_g \cdot prob$$

where the effect matching the condition of g has the probability *prob* to come true after the execution of k.

2. A behavior module k receives negative activation $a_{kg}^{t}\prime\prime$ from a goal g with importance I_g at activation cycle t if one of the effects of k is the opposite to one of the conditions of g:

$$a_{kg}^{t}\prime\prime = -\delta \cdot I_g \cdot prob$$

where the effect negating the condition of g has the probability *prob* to come true after the execution of k.

3. Let g be a goal and k and j be two behavior modules such that k has an effect with the probability *prob* that is one of the preconditions of j. Let $\tau(p_j, s)$ be the value of the state that is the precondition of j and the effect of k. Then the activation $a_{kg}^{t}\prime\prime\prime$ given from j to k is defined as:

$$a_{kg}^{t}\prime\prime\prime = \gamma \cdot \sigma(a_{jg}^{t-1}) \cdot prob \cdot (1 - \tau(p_j, s))$$

where

$$\sigma(x) = \frac{1}{1 + e^{B(\mu - x)}}$$

$\sigma(x)$ is an S-shaped Sigmoid filter used to filter the activation values so that large values of activation become larger and small activation values become smaller. Using a Sigmoid filter for this purpose was first proposed by Goetz and Walters [GW97] as a way to avoid frequent switching between behaviors. Including $(1 - \tau(p_j, s))$ in the calculation of $a_{kg}^{t}\prime\prime\prime$ implies that the less fulfilled a precondition is, the more activation will be given to behavior modules that fulfill that precondition. Note that there can be several behavior modules j:s that fulfill the precondition of behavior module k but $a_{kg}^{t}\prime\prime\prime$ is set to the absolute maximum value out of the activations of all the j-k pairs. Only the strongest link between a behavior module and a goal is maintained.

4. Let g be a goal and k and j be two behavior modules such that k has an effect with the probability *prob* that is the opposite of one of the preconditions of j. Then the activation $a_{kg}^{t}\prime\prime\prime\prime$ given from j to k is defined as:

$$a_{kg}^{t}\prime\prime\prime\prime = -\delta \cdot \sigma(a_{jg}^{t-1}) \cdot prob \cdot \tau(p_j, s)$$

The final activation a^t_{kg} given to the behavior module k by the goal g at activation cycle t is set to the activation that has the highest absolute value:

$$a^t_{kg} = absmax(a^t_{kg}\prime, a^t_{kg}\prime\prime, a^t_{kg}\prime\prime\prime, a^t_{kg}\prime\prime\prime\prime)$$

This implies that only the strongest path from each goal to a behavior module is used. Combining activations from the different links is not allowed.

The total activation a^t_k for behavior module k is the sum of the activations from all goals, with an addition of a part of the previous total activation, a^{t-1}_k for k:

$$a^t_k = \beta a^{t-1}_k + \sum_g a^t_{kg} \tag{2.1}$$

Action Selection

In each cycle the following procedure for action selection takes place:

1. Calculate the total activation a^t_k for each behavior module k.

2. Calculate the *executability* e_k for each behavior module k. The executability of a behavior module is calculated as the conjunction of the preconditions to the behavior module. The executability is a measure of how likely it is that the behavior can execute successfully.

3. Calculate the execution-value h as a combination of a^t_k and e_k by the use of a non-decreasing function, for instance:

$$h = a^t_k \cdot e_k \tag{2.2}$$

   Note that it is necessary to design the combining function in such a way that $h = 0$ when $e = 0$ to assure that no non-executable behavior modules are selected for execution.

4. For each resource required, check if the execution value, h, exceeds the local activation threshold for the resource. For each behavior module, check each required resource for availability and bind necessary resources. Once all resources exceed the threshold, the behavior module is chosen for execution and the bound resources are released. If no behavior module can be selected for a resource, the local activation threshold is lowered by $\Delta\theta$ and the procedure is repeated.

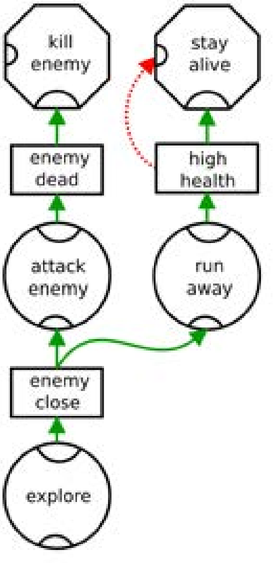


Figure 2.4: An example of a behavior network.

Example

An example of a behavior network is displayed in Figure 2.4. In this example, the agent is a simple NPC-fighter in a game. In general, the behavior of the NPC is the following. When there is no enemy in sight, the NPC will explore the environment. If there is an enemy close by the NPC will attack, unless its health is too low in which case it will run away instead.

In this example activation is spread in the following way. Assuming all behavior modules start out with activation value zero, the first cycle will propagate activation from the goal "kill enemy" to "attack enemy" and from the goal "stay alive" to "run away". During the next cycle, the same propagation is done, but furthermore there is activation spreading from "attack enemy" to "explore" and from "run away" to "explore". The spreading from behavior module to behavior module relies on the fulfillment of the precondition state. In this case, no activation will be spread to "explore" if the state "enemy close" is already true. This is to avoid the triggering of unnecessary behaviors. The state "high health" is a negative dynamic importance for the goal "stay alive". This means that if the health of the NPC is high, the goal to stay alive will not be as important.

Note that during the first cycle, activation only reaches the behavior modules that fulfill a goal directly. In this case, these behavior modules are "attack enemy" and "run away". It takes a few cycles for activation to reach the entire network, depending on the design of the network. Cycles can be performed repeatedly within one single agent update cycle until a behavior has been chosen, or it can be a continuous process where one cycle is performed per agent update. In the latter case, the planning properties of the behavior network will have a slight delay but reactivity will remain fast.

2.2 Pathfinding

One fundamental ability of characters situated in virtual worlds is to move from an initial position to a target position without colliding with obstacles on the way, choosing a reasonable path and not making strange turns when it is unnecessary. Despite having been studied for a long time, the problem of pathfinding is not solved with respect to realism and believability. There exist many methods that find the shortest path through an environment, but such a path may appear unnatural. In computer games or other virtual worlds, where realism and believability are important, natural-looking paths are an important aspect. Until now, many pathfinding techniques have focused on finding the shortest paths, omitting to address more complex forms of pathfinding.

There exist a large variety of pathfinding methods both in academia and in the gaming industry. Some algorithms worth mentioning are the Focussed D*-algorithm [Ste95], potential fields [Kha86], visibility graphs [LPW79], navigation meshes [LD04, KBT03], and the Corridor Map Method (CMM) [GO07, GS10, OKG08]. These were not used during the work in this thesis however. In this following sections, the algorithms used for pathfinding in this thesis work are briefly described.

2.2.1 The A* Algorithm

One of the first things a reader encounters when reading about pathfinding is the A*-algorithm. The A*-algorithm itself has nothing to do with pathfinding, per se. It is a graph search algorithm. The A*-algorithm performs its search on a *directed non-negative weighted graph* [Mil06]. An example of such a graph is shown in Figure 2.5. Given a start node and end node, the algorithm is guaranteed to find the optimal path through the graph, if such a path exists. The A*-algorithm uses the notion of *heuristics*, an estimation function that estimates the remaining cost from the current node to the goal node. This function must be optimistic, i.e. it must never give a higher cost value than the actual cost value, for optimal paths to be guaranteed.

The A*-algorithm performs a best-first search, keeping track of the costs of the nodes already visited. At each iteration, the node with the least total cost is expanded. The total cost $g(n)$ at node n is defined as

$$g(n) = c(n) + h(n)$$

where $c(n)$ is the cost from the start node to node n, and $h(n)$ is the heuristic estimate from node n to the goal node. If the heuristic estimate is good, the A*-algorithm will only traverse a small amount of nodes which are not part of the final path. This is what makes the A*-algorithm so efficient. Omitting the heuristics, the A*-algorithm is equal to the

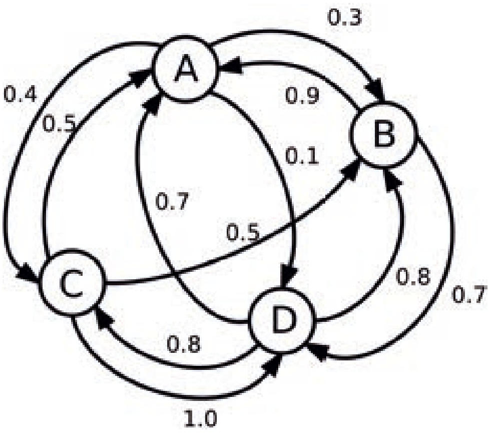


Figure 2.5: A directed non-negative weighted graph.

Djikstra algorithm [Dij59]. Such a search algorithm takes a longer time to execute because it traverses more nodes that are not part of the final part. For pathfinding, the Euclidean distance is often used as the heuristic function. This works well if the environment consists of large open spaces, but works less well when there are many obstacles and corridors.

2.2.2 Local Multiresolution Pathfinding

Behnke suggests a multi-resolution approach to pathfinding (with an emphasis on pathfinding for robotics) [Beh04]. The method is meant to be used when the environment is not fully known or there is uncertainty in where obstacles are. The environment is represented by a regular two-dimensional multi-resolution grid structure. The resolution of the grid is set higher close to the current position of the robot. Obstacles are represented as increasingly large, but decreasingly costly, disks with respect to the distance from the robot. The idea is that if the obstacles are far away, the knowledge about them may not be correct. Therefore, the cost, which would normally fully prevent the robot from moving into the area, is spread out over a larger area. The pathfinding is updated when the robot has changed position, to ensure that the path close to the robot always remains accurate.

2.3 Emotions

The work in this thesis uses the concept of emotions to a great extent. While the work does not claim to actually give virtual agents real emotions, or simulate real human emotions, the background theory concerning emotions is used as a source for inspiration. In this section, there will be a general description of emotions and their impact on decision making. The majority of the psychological theories presented in this section are used extensively in the contributions of this thesis.

2.3.1 What Are Emotions?

It is sometimes relevant to define the difference between emotions, feelings, mood and personality. Generally, emotions are short-term, intense states that are context-specific. Feelings are usually defined as emotions that one has become cognitively aware of. Moods are more long-term, have low intensity, and are not connected to a specific event. Personality traits are extremely long-term, usually staying fairly constant over the course of a person's life. The term affect or affective states is a general term used to describe emotions, feelings, moods, and related bodily states.

There is great debate on what defines something as an emotion. What makes fear an emotion, but thirst something else? Traditionally, emotions have been viewed as a homogeneous category, but lately there has been evidence to suggest that the term emotions is not a category based on nature, but rather a category created by the human mind [Bar06, PB08]. From a neuroscientific point of view, the things we call emotions have little in common. Barrett [Bar06] claims that emotions are not a category which has physical support, and suggests a new paradigm where emotions are not given the explanatory power they currently hold in research. While many theories [Sch09] concerning emotions disagree to a certain extent with Barrett's ideas, the questions raised by Barrett's work nevertheless illustrate the discord in the research community when it comes to emotions. Furthermore, for most humans, the category of emotions seems natural and well-established despite the lack of neurological evidence.

For a long time valence, the "goodness" of an emotion, has been considered very important. Emotions have been divided into positive and negative emotions. It has been suggested, however, that valence is not only culture-dependent, but fails to describe emotions properly [SS02]. Nevertheless, many researchers still use the term negative emotions when addressing such emotions as fear or anger, and positive emotions when addressing e.g. happiness.

Another question highly debated among emotion researchers is the number of basic emotions. While some [YRFB99] claim that only valence, together with arousal, is needed, resulting in a good vs. bad categorization of emotions, others suggest that four dimensions are needed to capture the essence of the emotions [FSRE07]. Others again claim that there could potentially be a near infinite number of emotions [Bar06, Fri88]. Nevertheless, among the numerous emotion models that exist, some emotions are more often classified as basic than others. *Fear*, *anger*, *joy*, and *sadness* are most often mentioned (for a good review of emotional theories, see paper by Nesse and Ellsworth [NE09]).

Paul Ekman is known for his work on emotions and their related facial expressions [Ekm92, Ekm99]. He presents evidence that the expressions of emotions are independent of cultural background and hence are likely a result of evolution rather than a social construction.

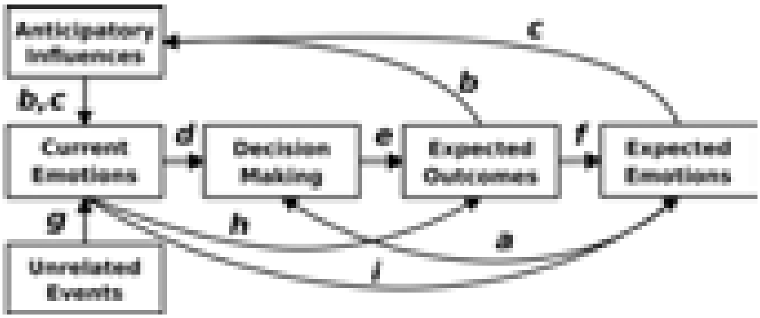


Figure 2.6: The Loewenstein and Lerner model for emotional decision making.

2.3.2 The Emotional Impact on Decision Making

Emotions have been studied in many different areas, e.g. psychology, neurology, and economy. Damasio is well-known for his work [Dam95, Dam99] discovering the importance of emotions in decision making from a neurological point of view. Damasio and his colleagues have studied several cases where patients had injured their frontal lobe. This type of brain injury differs greatly from other types of brain injuries in that the patients often show no impairment in intellect, communication, problem solving, long-term memory or working memory [BDDA94]. Nevertheless, many of them have grave problems leading normal lives and making everyday decisions. It becomes clear that these patients lack the ability to make proper decisions in everyday situations. What is also evident in these patients is an impairment in their emotional state. Damasio suggests in his work [Dam95, Dam99] the *somatic marker* hypothesis, a way for the mind to attach emotions to the alternatives one currently must choose between. In a real-life decision-making situation, relying on reasoning alone will not be enough since there are usually uncertainties or incomplete knowledge concerning the different aspects of the selectable alternatives. Furthermore, there are often numerous alternative options to choose between, making it difficult to completely evaluate all possible options. Damasio believes that emotions are vital to refrain the brain from trying to solve all decisions by pure reasoning/logic alone. In a sense, emotions work as filters to speed up and enhance the decision-making process. Furthermore, it is known that humans who lack emotional capabilities have problems deciding even between few, well-defined alternatives.

Within the fields of psychology and economics, Loewenstein et al. have done extensive work on emotional decision making [Loe96, LWHW01, LL03]. Loewenstein and Lerner suggest a general model for how emotions influence decision making [LL03]. Their model is depicted in Figure 2.6. They suggest that there are two ways emotions influence decision making. The first group, *expected emotions* are emotions that the person believes will be the result of the various choices he/she makes. It is presumed that humans evaluate their decisions in terms of the emotional consequences these decisions will lead to. In general, humans try to maximize their positive emotions while minimizing their negative emotions.

To do so, all alternative actions must be evaluated in terms of their possible emotional effects (link a in figure). It is worth mentioning that the effects can be long-term. For instance, we choose to get an education now, although it requires a lot of effort, because we believe that it will make us happy later on in life (better job, higher salaries, more job fulfillment). The second group, *current emotions*, are emotions that the person is feeling at the time the decision making takes place. These emotions can be completely unrelated to the decision making task at hand (link g). For instance, being angry because your car broke down might affect how you treat people at work, even though the decisions at work have nothing to do with your car. The current emotions, however, can also be due to *anticipatory influences*, which are a result of the decision-making process itself. When a person tries to make a decision he/she analyzes the different alternatives and weighs the different benefits and drawbacks against each other. This process itself can give rise to emotions (link b and c). For instance, while considering taking an action that could result in something rather dangerous, emotions such as fear may be triggered by just considering that action as an alternative. Current emotions affect the decision making in two ways: *directly* and *indirectly*. Emotions directly affect individual decisions, trying to answer the question "How do I feel about this alternative?". Moreover, emotions affect the reasoning about the outcomes of the potential actions (link h) and the possible emotions invoked by that action (link i). Emotions indirectly affect the actual decision-making process, for instance, how far ahead the person can plan, and how big a risk is viewed as acceptable.

While humans try to (consciously or subconsciously) maximize their positive emotions while minimizing their negative emotions, humans in general perform poorly at forecasting their future emotions [WG03]. Because of this, we make choices based on what we *believe* we will feel, not what we actually will feel. It is known that humans are very poor at knowing what makes them happy [HH06]. Sometimes the reason behind our poor decisions is incorrect predictions about future events or incorrect memories from the past. However, we can make poor choices even when we recognize that they are poor (e.g. smokers continue to smoke despite being fully aware of the dangers involved).

Schwarz and Clore have conducted a series of studies on what they name *feelings-as-information* [SC83, Sch00, SC03, SC07]. They have performed mood-inducing experiments with the attempt to explore how emotions and moods affect the perception of events. It was established that emotions, especially sadness, need to be explained to the human mind in some way. For instance, when a person feels depressed because of bad weather, but is not aware that the bad weather is the source of the bad mood, the person says that he/she is not so satisfied with his/her life. However, if the person's attention is brought to the source of the bad mood, in this case the weather, the impact of that mood becomes negligible and the person can give a more accurate estimation of his/her life satisfaction. This suggests that if we are unaware of the source of our emotions, they will affect how we view the world around us, affecting our decisions.

Isen et al. have conducted studies on how positive affect influences decision making [IJMR85, IDN87]. They have concluded that positive affect improves creativity related to problem-solving. A person in a good mood sorts and processes ideas differently from a person in a bad mood. A person in a good mood performs better at tasks that involve creativity.

It is clear that emotions are vital for human decisions. A lack of emotional attachment and response to the consequences of our actions is evident in society [Loe10]. An average person in a wealthy country causes a substantial amount of damage to people in developing countries by indirect means through the support of poor employment conditions, the use of dangerous chemicals at factories, etc. While people are cognitively aware of the damages caused by their actions their choices change little, because of the non-specific nature of that knowledge. We fail to empathize with the people we indirectly harm because we do not feel an emotional attachment to them. When emotional messages are given to us, portraying the same knowledge we already have, we are much more inclined to change our behavior. It is the emotions that drive us rather than pure facts.

Memory

LeDoux is known for his work [LeD96, LeD00, LeD03] studying the neurological backgrounds for emotions. The work on how emotions affect memory, in particular, is of relevance for this thesis. His studies often focus on fear-conditioning. When an animal is exposed to unpleasant stimuli whenever it encounters a neutral context, it will after some time develop a fear for that context. LeDoux shows that memories are more easily retrieved when a person is in a similar emotional state to the one the person was in when the remembered event was experienced. This is highly important for all types of decision making, since it greatly influences the knowledge and memories available to the decider.

Risk

How humans perceive risk is strongly affected by their current emotional state. Studies by Lerner and Keltner [LK00, LK01] have shown that although anger and fear are both considered negative emotions, they have opposite effects on risk perception. People who are angry are optimistic in their prediction of possible risks involved in the decision, while fearful people are pessimistic. Their studies show that in this respect anger is more similar to happiness, an emotion which also promotes optimism, than to fear. Similarly, a study conducted by Raghunathan and Pham [RP99] proposes that different negative emotions do not have similar effects on decision making. While anxiety is clearly connected to risk-avoidance, sadness permits greater risks and has a large emphasis on high rewards.

Risk is also greatly affected by how we feel about the concept connected to the risk [SPFM05]. When we like something, we will correspondingly lower the perceived risk for that concept and vice versa. When we make a choice it may be very unlikely that the thing we fear will be a result of that choice. Nevertheless, our fear of the outcome may overrule our sense of rationality. An obvious example is the fear of flying. While statistically one of the safer ways to travel, many people feel intense fear related to the risk of a plane crash. The idea that our emotions and feelings towards the object of the risk affect our perception of that risk is called *affect heuristics*.

Humans have difficulties distinguishing between risks, or probabilities, that are really small in magnitude. For instance, there seems to be little difference in how a person reacts emotionally to a risk that has a 1 in 100,000 chance compared to a 1 in 100,000,000 chance of occurring [LWHW01]. This naturally also affects how humans view risk probabilities in the decision-making process. Humans also perceive a risk as greater if it is presented in frequency scales (e.g. one person in 10) than if it is presented using probabilities (e.g. 10% chance) [SMM00]. The reason for this is assumed to be that frequency scales portray a sense of realism that gives rise to emotions related to the risks.

Time-Discounting

Time plays a great role in decision making. Often, when choosing a certain alternative, one is expecting a certain result. If this result is far in the future, or nearby, plays a great role in which course of action one will take. Generally, time is taken into consideration as a form of cost. This is called *time-discounting* [WP08].

The amount of time-discounting a person does is greatly influenced by his or her emotional state. People who are in an elevated emotional state (such as people who are very angry or afraid) are more impulsive and focus on fast rewards. They will tend to disfavor actions that involve delayed gratification [LL03]. The tendency for impulsive actions during distress is mainly due to the belief that one's mood will improve as a result of the instant gratifications [TBB01]. It is also suggested that emotional arousal affects an internal "pace-maker", making time seem to move slower for impulsive people [WP08].

Planning and Processing

Studies by Bless et al. [BBSS90] have shown that people in a good mood are more easily persuaded to change their opinion even when the arguments presented are not very strong. People in a bad mood, however, need to be presented with strong arguments to change their point of view. For an observer, this means that a person in a good mood may switch behavior without an apparently good reason.

Luce et al. [LBP97] have performed a series of tests to show that negative emotions lead to more extensive processing during decision making. Moreover, negative emotions induce focus on one attribute at a time during the decision-making process. This implies that people in a bad mood take longer to make decisions than people in a good mood.

Agent Architecture 3

The work presented in this thesis has to a large extent been related to the full-fledged implementation of an agent architecture. The purpose of the agent architecture is to control virtual characters for games or similar applications, including serious games and interactive installations. The author of this thesis is one of the main contributors to the development of the architecture. This chapter gives a general overview of the overall architecture. This is necessary to give a context to the work described in the thesis.

3.1 Architecture Overview

A general overview of the agent architecture is depicted in Figure 3.1. The agent architecture consists of several separate modules that communicate with each other. The blue box signifies the modules that are part of the agent architecture. Each update cycle new information is sent from the simulation engine through the communication interface to the agent. This information is first parsed by the internal perception and then sent to the knowledge base where it is stored in memory. The new information is also sent to the appraisal where it is evaluated in terms of its emotional relevance to the agent. The appraisal module uses this new information together with the personality of the agent, expectations, as well as old information to trigger emotion signals, which are stored in the emotion module. The emotion module is used by nearly all other modules, which is represented by the large, short arrows in the figure. The action selection makes high-level decisions on what actions to perform next. The result of these decisions is sent to the action management, which breaks the abstract actions down into more manageable actions if necessary. The action management outputs the basic actions to the communication interface, which in turn sends them to the simulation engine to be executed. Finally, when a decision is made in the action selection module expectations are triggered. The agent architecture is completely decoupled from the simulation engine used, making it easier to switch engines as desired.

In the next sections, the different modules of the agent architecture are briefly described. The chapter finishes with a description of a project, Animalistic, where this architecture has been used. In this project, the agent architecture is used to control virtual animals in

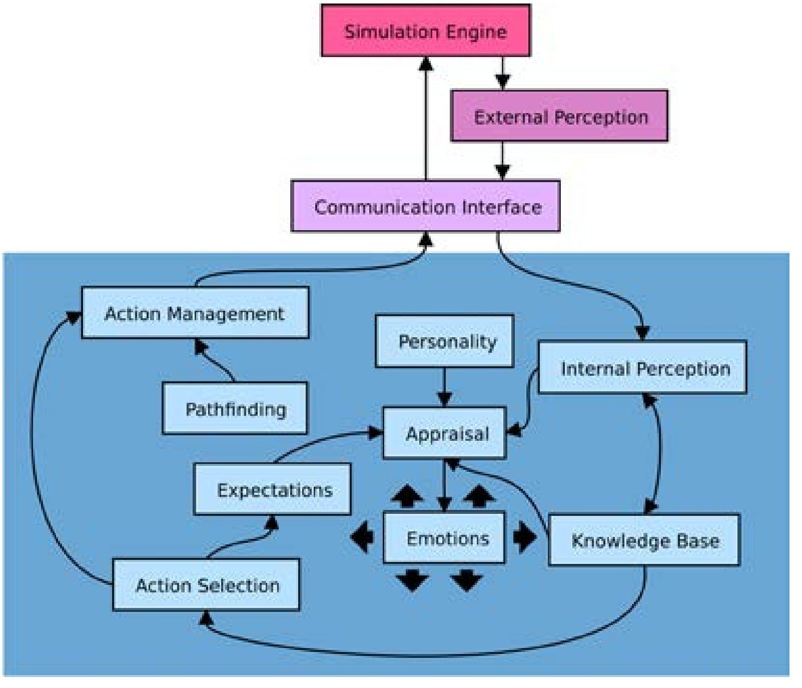


Figure 3.1: A schematic view of the agent architecture. The blue area represents the agent architecture, while the outlying boxes are part of the simulation.

an interactive installation.

3.2 Communication Interface

The different modules of the agent architecture are connected to each other, forming an *agent unit*. The agent unit can be viewed as the brain of the agent. It cannot perform any actions itself, but merely relays its wishes to the simulation engine. If possible, these requests will be performed. Communication in and out of the agent unit, as well as most of the communication between modules, is conducted using XML, due to suitable expressive properties of the language.

3.3 Emotion Module

In the agent architecture emotions and physiological states are represented as a sum of Sigmoid signals. Once a signal has been triggered, it stays in the emotion module until it

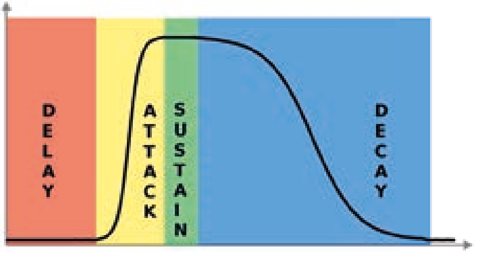


Figure 3.2: The four phases of the emotion signal: delay, attack, sustain, and decay.

has decayed completely, then it is removed. The signals are described in detail below.

3.3.1 Sigmoid Signals

A signal consists of four parts: the *delay* phase, the *attack* phase, the *sustain* phase and the *decay* phase (see Figure 3.2). A signal also has an intensity value which lies in the interval $[-1, 1]$. Note that this implies that an emotion signal can have a negative intensity. This is usually a result of the emotion correlation system (see description below). The emotional value $signal(t)$, at time t after the emotion has been triggered, can be calculated in the following way. Note that t is normalized to lie in the interval $[0, 1]$ for each of the phases in the formula below.

During the delay phase the signal has no value:

$$signal(t) = 0$$

During the attack phase the signal starts to grow in strength:

$$signal(t) = \frac{I}{1 + e^{-(t-h)/s}}$$

where I is the maximum intensity of the signal. The parameters h and s are used to define the shape of the Sigmoid function. The max intensity of the signal is maintained during the sustain phase:

$$signal(t) = I$$

During the decay phase the signal decreases in value.

$$signal(t) = I - \frac{I}{1 + e^{-(t-h)/s}}$$

The emotional value *emotion* for a particular emotional state (e.g. fear) is calculated as the sum of all N signals of that type, clamped to lie between 0 and 1.

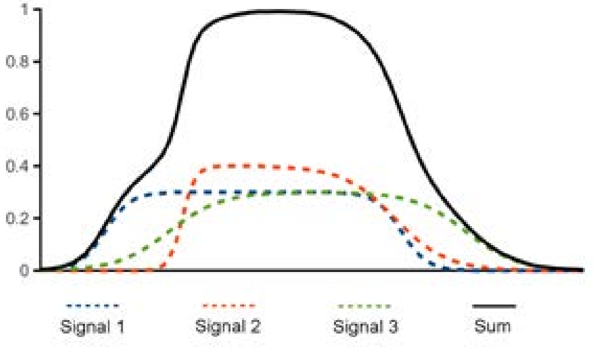


Figure 3.3: An example of how emotion signals are combined into an emotion value.

$$emotion = min(max(\sum_{i=1}^{N} signal_i, 0), 1)$$

The summing up of the signals is shown in Figure 3.3. In this example, there exist three signals for the emotion in the emotion module. The value of these signals is shown as functions of time. The summed up values for the emotion these signals represent is shown in black.

3.3.2 Emotion Correlation System

While it is difficult, perhaps impossible, to specify which emotions are opposites of each other, the emotion module supports an automatic system for triggering emotions in correlation to other emotions. For instance, if something in the environment triggers a fear signal, positive emotions such as happiness and pride should be lowered automatically. For each emotion in the system, the emotion correlation system allows the designer to specify a set of emotions coupled with intensity constants. Assume the following configurations are made:

```
fear
   happiness -1
   anger -0.8
```

Whenever fear is triggered with intensity I, two signals representing the emotions happiness and anger are also triggered. The intensity of the happiness signal is $-1 \cdot I$ and the intensity of the anger signal is $-0.8 \cdot I$. The lengths of the delay, attack, sustain, and decay phases are identical to those of the fear signal.

3.3.3 Filtering System

The emotion module also includes a filtering system, where emotions affect each other. The filtering system is similar to that described by Picard [Pic97] and the DER system implemented by Tanguy et al. [TWB07]. All the parameters of an emotional signal (intensity, delay, attack, sustain and decay) can be influenced by the values of other emotions. This is highly useful, as it can mimic how emotions such as fear strongly inhibit the exciting of other emotions. Currently, three types of filters exist: sigmoid filters, gamma filters and linear filters. Using these filters, it is possible to design a variety of emotional interactions.

The difference between the filtering system and the emotion correlation system is the following. The emotion correlation system takes an input emotion signal and triggers new emotion signals according to some rules. It does *not* change the original input signal. The filtering system, on the contrary, takes an input emotion signal and uses already existing emotions to change this input signal before it is put into the signal memory in the emotion module.

3.4 Appraisal

The affective appraisal module triggers emotions depending on what happens in the virtual environment. Currently the appraisal triggers emotions in two ways. First, it has a set of simple rules that trigger specific emotions with certain strengths, e.g. "if a wolf is seen, trigger fear". The actions selected in the behavior module also creates expectations that are evaluated by the appraisal and the appraisal module triggers emotions during these evaluations. Other parts of the appraisal module are under development.

3.5 Perception

Perception is divided into two parts. First, there are physical limits (such as distance to event and the field of view) to what the agent can perceive. These are factors that are beyond the control of the agent. This perception is done outside of the agent unit and depends on the game engine used (*External Perception* in Figure 3.5). The second part of perception has to do with how the agent manages the information that is picked up by its senses. The *Internal Perception* module manages the incoming data and attaches importance to the different perceptions depending on how important they are to the agent. Currently this is done by searching for a given set of keywords in the incoming data and adjusting the importance of this data accordingly. An example of such a keyword configuration is seen below.

```
importance 0.1
   type value="box"
   color value="blue"
```

The keyword above states that if a blue box is found, the importance of that perception should be 0.1. If a perception matches several keywords, the importance values are added. The importance is later used in the memory module to define how long a memory is remembered.

3.6 Memory

The memory module, or *knowledge base*, works as a database that can be queried by other modules to extract information. The memory module allows the character to remember information over time as well as attach emotional attributes to each memory. The emotions attached to a memory correspond to the emotions of the agent at the time the memory was acquired. Each memory can be composed of any information represented as a structure of tags with attributes and subtags. A typical memory has the following structure:

```
object id=3
   type value=wolf
   position x=1 y=5 z=7
   holds_item id=9
```

Stored along with this memory is the time at which it was acquired by the agent, the importance of the memory (set by the perception module), and the emotional state at the time of acquisition. Forgetting works by using the importance of each memory as a variable in the forgetting curve [Ebb15], which determines how long a memory is remembered. Currently the memory module either remembers something completely or it deletes it from the memory forever. It is not possible for the agent to momentarily forget about something only to remember it later.

3.7 Personality

The personality module is based on the OCEAN model (also known as the NEO-PI model) [CM92] and has five different attributes: *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism*. Openness describes creativity, flexibility, curiosity, etc. A high level of conscientiousness describes a person who is meticulous, well-organized, and hard-working. Extraversion expresses how social a person is, how active and how easy it is for the person to feel positive emotions. A person with a high level of agreeableness is cooperative, trusting and sympathetic. Neuroticism describes a person's proneness to feelings of great psychological distress.

The personality module itself only provides information about the personality of the character. It can be used wherever appropriate, for instance in the decision-making or in the affective appraisal. Currently, however, it is only used in parts of the appraisal module.

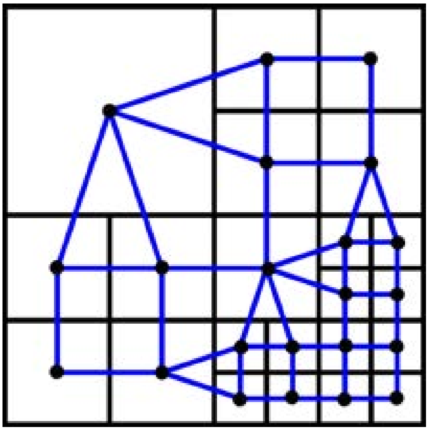


Figure 3.4: The hierarchical structure for the pathfinding.

3.8 Action Selection

The action selection module performs the high-level decision making. This module currently supports two types of action selection mechanisms; behavior networks and behavior trees, both of which are described in more detail in Chapters 2 and 4.

3.9 Action Management

The action management module receives the output from the action selection module and can, if necessary, break down the high-level actions into low-level actions or perform computational tasks, such as pathfinding. For example, if the action selection outputs a high-level action such as “get money from the bank”, the action management module breaks this abstract action into smaller, manageable actions such as “put on shoes” followed by “walk to bank” and “withdraw money”.

3.10 Pathfinding

Pathfinding is performed using the A*-algorithm on a hierarchical grid structure similar to the one suggested by Behnke [Beh04]. Figure 3.4 shows a 4-connectivity grid where the blue lines are the edges in the graph submitted to the A*-algorithm. The resolution of the grid is higher close to the agent and to obstacles. The cost of the A*-algorithm is calculated using a configurable set of influence maps.

3.10.1 Influence Maps

The influence map manager creates, updates and contains a set of *influence maps* [MF09, Toz01]. There is no formal definition of influence maps, but in the agent architecture they are 2D environmental maps that are used to assist the character's decisions. These influence maps can be merged by the game designer in different ways using easy-to-use configuration files. The influence maps are often used by the pathfinding module, but can also be used for strategic decisions. Currently, there are several different types of influence maps, including distance maps, emotion maps, visibility maps, probability maps, maps of visited areas, and object maps. Emotion maps and visibility maps are part of the thesis contributions and are described in detail in Chapter 4.

3.11 The Animalistic Project

The agent architecture was used as a part of the Animalistic project at Norrköpings Visualiseringscenter in 2011. The Animalistic project was a half a year long interactive installation at the center. The project made use of the Kinect device to allow visitors to the center to interact with virtual animals in a virtual world. The installation was developed mainly for children, but any visitor could use it.

The animals in the virtual world were extra-terrestrial in nature, inhabiting a vast field of grass. The animals could, depending on their mood, decide to sleep, eat, explore the world or walk up to the screen to interact with the visitors. There were three different species of animals in the installation. Each had a different type of behavior, such as a preferred resting place.

The form of interaction was, due to limited resources, simple. Standing in front of the screen attracted animals unless too many people were present, in which case the animals were frightened.

In the installation, the following affective and physiological state were used: hunger, happiness, fear, and socialness (the current need to socialize with a human). While the animals were unable to display emotions in a sophisticated way (e.g. through blended animations of different affect type), they were able to portray their current thoughts and feelings to the visitors through thought bubbles appearing over their heads. This addition was made partway through the project when it had become evident that the reason behind the characters' actions was not clear to the visitors.

A user study was planned and executed during the Animalistic project. The study was intended to answer the following questions. First, did the animations of the virtual animals convey the desired behavior to the visitors? Secondly, what were the movements used by the users when they were asked to interact with the animals? A school class of pupils at around age 11 volunteered for the user study. The first part of the study was executed using a questionnaire. The second part used video recordings of the children as they interacted with the installation. The first question was answered partly during the study, resulting in a slight change of the animations for final installation. Results for the

Figure 3.5: A child interacting with the virtual animals in the installation. Photo courtesy of Norrköpings Visualiseringscenter.

user interaction part gave few results, as most children did not to interact with the animals at all.

The Animalistic project gave a much needed opportunity to try out the agent architecture in a real application. The results from the project showed that one of the main problems with EmoBNs is the difficulty in tuning the different variables and parameters of the network. The parameters $\beta, \delta, \gamma, \theta$ and $\Delta\theta$[1] are fairly straightforward to configure. The real problem lies in configuring the probabilities and time-intervals for each behavior effect. A small change in probability sometimes greatly affects the frequency of a behavior. The biggest problem lies in the debugging. It is often difficult to notice errors in the configurations of the EmoBNs because of the non-hierarchical structure of the network.

[1]These parameters control the overall activation spreading in the behavior network. For more information, see Section 2.1.4.)

CONTRIBUTIONS 4

This chapter presents the contributions of this thesis. It is divided into three parts outlining the contributions in believability, emotional decision making, and pathfinding.

4.1 BELIEVABILITY

This section presents the first contribution of the thesis, namely the work that concerns believability in gameplay, specifically determining the properties that an agent architecture needs to achieve believability. This work is described in detail in Paper B.

Believability is vital for non-player characters in computer games. Without believability, the user experience is greatly diminished. Lankoski and Björk have analyzed the reason behind believability in games [LB07b, LB07a, LB08]. In Paper B the gameplay design patterns proposed by Lankoski and Björk are used to evaluate the possibility of designing believable characters using an agent architecture that was under development at the time. Below, this architecture is referred to as AA1.

4.1.1 DESIGN PATTERNS FOR NPCS

Lankoski and Björk define a set of gameplay design patterns that can be used for analyzing believability in NPCs. The gameplay design patterns are described as "semiformal interdependent descriptions of commonly reoccurring parts of the design of a game that concerns gameplay" [BH05].

Note that the concept of gameplay design patterns is not part of Paper B nor the thesis contribution. The goal of the paper was to use gameplay design patterns to analyze AA1 with the aim of identifying weaknesses and the need for further extensions. The gameplay design patterns are described briefly below.

Awareness of Surroundings The ability to detect and react to all relevant events in a believable way. An example of this is escaping when an enemy attacks.

Memory of Important Events The ability to remember past events that were of relevance to the NPC. This is important for NPCs with a rich personality. For instance, if the player has stolen chickens out of the NPC-farmer's yard several times before, the farmer should have a negative/watchful attitude towards the player.

Emotional Attachment The way a character feels emotionally about different people, objects or events. This can encompass everything from fierce phobias for snakes to simply preferring action movies over comedies.

Sense of Self The ability to notice events and respond to these according to the NPC's own interests, goals and current internal states. If an NPC is picking turnips and the player comes along and "steals" a turnip from under their nose, it should recognize that this behavior goes against its wishes and respond by showing dissatisfaction towards the player.

Initiative The ability to take actions that are not merely a direct perceivable reaction to a current event. For instance, an NPC that uses a shield when someone attacks it does not fulfill this pattern, but an NPC that sneaks up to the enemy camp to steal ammunition does.

Own Agenda The character seems to strive towards its own personal goals and interest, rather than just simply follow orders. For example, the actions of a merchant NPC could show that its goal is to profit from its transactions and avoid robbery.

Goal-Driven Personal Development The ability to update goals when an old goal is fulfilled or cannot ever be achieved. These new goals must be plausible from a psychological and narrative point of view. If a merchant NPC has had the goal to cheat a customer to make money and this goal fails, it should not suddenly switch to doing charitable actions.

Unpredictable Behavior The ability to perform actions that are unexpected but nevertheless fit the context and the character's personality. An NPC fighter that suddenly aborts its attack on the player, to jump behind a big rock is feasible, but an NPC fighter that aborts its attack only to pick flowers is not.

Open Destiny The character can have different narrative arcs, have different destinies, between different game instances. Many players play the same game several times and if NPCs behave in the same way every time this makes them less interesting.

4.1.2 Agent Architecture

To a large extent, the AA1 architecture described in Paper B is similar to the architecture described in Chapter 3 (this architecture is called AA2 in the text below). There are some differences, however. The learning module is mentioned in the paper. Learning for behavior networks was indeed implemented, nevertheless the results were not satisfying enough for usage in the AA2 architecture. Similarly, the advanced visual perception never made it to the final implementation of the AA2 architecture.

The major contribution of Paper B is to analyze which of the above design patterns the AA1 architecture can fulfill, and how. Since the architecture has changed somewhat since the paper was written, the following section contains some modifications to the original results. In general, it is difficult to use gameplay design patterns to analyze an agent architecture in terms of believability properties. For instance, there is a great difference between, for example the appearance of having goals and having actual goals in the decision-making system. Therefore, it is difficult to determine if e.g. the design pattern *goal-driven personal development* is fulfilled. The appearance of goals does not necessarily require real goals in the architecture, but one can assume that it is easier to design behaviors that give the impression of goals if one has the means to express these goals in the architecture. The analysis is modified slightly from the one in the paper to reflect this problem.

Next is a description of the contributions of this work with respect to the design patterns mentioned in Section 4.1.1.

Awareness of Surroundings

The perception module of the agent architecture allows assigning importance values to objects and events. The preferences of each virtual character can be different. The character can then react to events or objects using its own preferences. However, while the architecture gives each incoming perception an importance value, this does not imply that the character will actually react to this object. To achieve a plausible reaction, which is the key for *Awareness of Surroundings*, the actual actions must be implemented in the decision-making module. Using the emotional properties of EmoBNs (described in detail in Section 4.2.1 and 4.2.2) can make the reactions of the character more plausible, because the player can more easily relate to emotional behavior.

Memory of Important Events

The memory module in the architecture as described in the paper as well as in Chapter 2 can remember memories of both objects and events. Depending on the importance value of a memory, it can be remembered longer. Therefore, memories about important events can be accessed at any time during the character's "life". To be able to display a behavior that shows that the character can remember old events, these types of behaviors must be incorporated into the behavior networks. This is possible in the proposed model, but it is important to note that the proper behavior only occurs if care is taken to include all necessary information in the behavior networks. The behavior network needs to know what behaviors to perform depending on what has happened previously.

Emotional Attachment

Using the AA1 architecture it is possible to implement emotional attachment. The perception module can use a list of preferences, assigning more importance to preferred objects or events. The affective appraisal can also trigger emotions accordingly. But merely triggering emotions is not enough. To be able to fulfill the emotional attachment pattern, the character must have a behavior that reflects its attachments. It is possible to do this with the behavior network but the game designer must remember to do so for all relevant characters, objects, and events.

Sense of Self

The sense of self pattern is possible to fulfill using the AA1 architecture. Nevertheless, although the character has its own interests and goals, it cannot automatically know if something around it has caused an undesirable change. Currently the architecture does not support automatic appraisal of events according to the agent's goals and interests. However, the sense of self pattern can be implemented by manually encoding such behavior into the behavior networks. For instance, it is possible to tell the character to show disgust if someone steals one of its items.

Initiative

It is possible to create both reactive and planning behavior using the emotional behavior network. The planning is especially important to fulfill the initiative pattern. Planning allows the character to think further ahead to fulfill its own goals and intentions. This in turn gives the illusion that the character has a mind of its own and does not merely react instinctively to what is currently happening in its surroundings.

Own Agenda

Using the proposed AA1 agent architecture, the character *will* have its own agenda. In fact, behavior networks require that the character has its own set of goals and ways to achieve those goals. A goal in behavior networks can also have a dynamic importance, which will make a goal become more important in certain circumstances. It is still not necessarily trivial to portray this agenda to the player. This must be done by creating a rather obvious behavior that accompanies the character's goals in life. For example, if the character has a goal to be rich this could be reflected in the dialogue, in its response to getting money, and in its way of making deals with the player.

Goal-Driven Personal Development

Behavior networks are highly suitable to fulfill the goal-driven personal development pattern. While the overall goals (e.g. to be happy) never change in the emotional behavior network, the subgoals (e.g. to gather food) are the ones that are most apparent to the player. The environment is constantly checked to see whether it is possible to perform a behavior. Furthermore, emotions influence the activation spreading of the network. Hence the character will always choose to perform a behavior that feels like the best possible behavior given the present circumstances. Naturally, care must be taken when designing the behavior network so that all plausible ways to achieve a certain state are taken into account. For example, if the character has the goal to gather as much food as possible but does not appear to notice the big apple tree close by, the player will find this behavior implausible. Currently, an EmoBN does not make adjustments to the nodes or parameters of the network if the expected result of an action does not occur although the character has performed it multiple times. It is therefore important that the game designer manually encodes information into the behavior network concerning when to give up a particular course of action.

Unpredictable Behavior

Unpredictable behavior can be addressed using EmoBNs. Sometimes emotions will make the character do things that may or may not be reasonable. Perhaps it will choose a less risky alternative over a more risky one when it is afraid. This will result in a behavior that is not predictable but will make sense once it occurs. It is also possible to hand-code more specific situations into the behavior network, should one wish to do so.

Open Destiny

Open destiny is a difficult pattern to achieve. However, by using long-term emotions or by possibly even letting certain events make permanent changes to the agent's personality, different behaviors may occur. A large behavior network has to be designed to handle a wide variety in destinies. The same is true regardless of the decision mechanism used.

4.1.3 Discussion

Paper B also discusses possible problems that are evident in commercial games but are not yet addressed by the proposed agent architecture. For instance, in many games (Oblivion, Skyrim, etc.) it is possible to steal items from the houses of the NPCs. While the NPCs react if they see the actual theft, the NPCs do not notice the absence of any objects in their house if they missed the actual theft. Reacting to missing objects is a complex problem, not easy to address. The absence of a small item should not be noticed unless it is of great importance, but the absence of many objects should definitely spur the interest of the NPC. Another issue discussed is the problem of maintaining control. Adding more complexity in a system makes it more difficult to predict and maintain the behavior of the characters. EmoBNs support rather complex decision making but this also makes it difficult to control the behavior of the character.

4.1.4 Summary of Contributions

The contribution of Paper B lies mainly in analyzing the proposed architecture with respect to its relevance for the fulfillment of the gameplay design patterns. The development of the actual gameplay design patterns are not included in the paper nor in this thesis work. The aim of the work in Paper B is also to identify the need for possible extensions to the existing architecture as well as to the already planned extensions.

4.2 Emotional Decision Making

This section presents the second contribution of the thesis, namely the research work carried out to introduce emotions into decision making mechanisms according to psychological theories, with the aim of creating a more believable behavior for virtual characters.

The first contribution is to introduce emotions into behavior networks. The second contribution is to extend the result of the first contribution to also include emotional time-discounting. The third contribution is to compare emotional behavior networks and behavior trees, focusing on differences in functionality and design. The result of this comparison shows the need of an extension to behavior trees. As a result, the fourth contribution is to introduce emotions into behavior trees. The work described in this section is presented in detail in Papers A, C, F, and G.

4.2.1 Emotional Behavior Networks

Emotions are vital for human decision making. They affect the way humans remember and process information. They also affect such factors as the amount of risk we are willing to take. In Paper A, EBNs (described in Section 2.1.4) are extended to include emotional impact. The inspiration for the work is the emotional model of Loewenstein and Lerner [LL03], as described in Section 2.3.2. The new extension is called *emotional behavior networks* (EmoBNs). The extensions to EBNs and the contributions to this thesis are described below, followed by the test results.

Emotional Parameters

The parameters of the behavior networks are $\gamma, \delta, \beta, \theta$, and $\Delta\theta$, as described in Section 2.1.4. In EmoBNs, the parameters of the behavior network are effected by the current emotional state of the agent. The parameters affect the activation spreading in different ways. The inertia parameter β affects how easily the agent switches behavior. A low value of β will result in a more flip-flop behavior, where the agent has a tendency to switch behaviors more often and more easily. The parameters δ and γ both affect how far in the future the agent can plan. Using low values for these parameters will give much less activation to behavior modules that are further away from goals. $\Delta\theta$ corresponds to the time it takes to make a decision. A lower value for this parameter means a longer time before decisions are made. Different emotions affect different parameters. The emotions used in Paper A can be seen in Table 4.1. The choice of emotions and how these emotions affect the parameters is based on theory presented in Section 2.3.2, but can be adjusted to better fit other theories if needed.

Emotional Probabilities

Each behavior module can have one or more effects, i.e. the consequences of the behavior. Each effect is coupled with a probability *prob*. Emotions affect how optimistic and pes-

Table 4.1: Affective impact on the behavior network parameters.

Parameter	**Negative**	**Positive**
γ	fear hunger anger	sadness fatigue
δ	fear hunger anger	sadness fatigue
β	happiness	anger sadness fear
$\Delta\theta$	anger sadness fear	happiness

simistic we are, and also how much risk we are willing to take (see Section 2.3.2 for full details). Therefore, it is suggested in Paper A that emotions should affect the probability of the effects. To do this, each effect must know the perceived "goodness" of the effect. If the character is optimistic, the probability for a negative effect should decrease while the probability for a positive effect should increase. The perceived goodness is called *benevolence* and is a user-specified value between -1 and 1, where -1 denotes a very negative effect and 1 denotes a very positive effect. The benevolence is specified for each effect of a behavior module. Using this value, the new emotional probability $prob_{emo}$ is calculated as follows:

$$prob_{emo} = prob \cdot (1 + sign(benevolence) \cdot (pos_{emo} - neg_{emo}) \cdot K)$$

where K is a constant used for determining to what extent emotions should alter the perceived probability, and pos_{emo} and neg_{emo} are the average values of emotions that affect risk-taking positively and negatively, respectively[1]. For example, happiness and anger make humans subjectively increase probabilities of favorable events occurring, while at the same time lowering the probabilities of negative events happening. In this case, the formula above changes the probability to a higher value if anger is high and benevolence is positive.

[1]It is suggested in Paper A to use happiness and anger for positive and fear for negative, but these can be changed if preferred or if other research results in psychology are discovered.

Emotional Influences

Emotions should be able to affect individual behavior modules directly. For instance, people might be more inclined to dance if they are happy than if they are sad, regardless of the effect of dancing. However, being happy is not a requirement for dancing and should therefore not be a precondition. Instead, it should just increase the already existing execution-value of the behavior module slightly. The work in Paper A extends behavior modules to include a set of *emotional influences*. Each influence is coupled with an emotional state and a *strength* that determines the extent to which the influence should affect the execution-value of the behavior module.

The total direct emotional influence Ψ on a behavior module with N influences is calculated as:

$$\Psi = \sum_{l=0}^{N} value_l \cdot strength_l$$

where $value_l$ is the value of the emotional state of influence l and $strength_l$ is the strength for influence l. Ψ is used to affect the calculation of the execution-value h (as defined in Equation 2.2 on page 15) in the following way:

$$h = a \cdot e \cdot (1 + \Psi)$$

Note that since the previous activation value (a^{t-1}) and not the previous execution-value (h^{t-1}) is used in the calculation of the current activation (see Eq. 2.1), emotional influences will only affect the behavior module locally. The change in the execution-value is not spread to other behavior modules in the next update cycle.

In the figures, the emotional influences are attached to the behavior modules through the left arc.

Emotional Goals

Psychological theories suggest that humans make decisions that try to maximize positive emotions while minimizing negative emotions (for more information, see Section 2.3.2). The ultimate reason behind many ordinary actions is to increase positive emotions, such as happiness, and likewise minimizing negative emotions, such as fear or sadness. This is evident when analyzing the reasons behind choices.

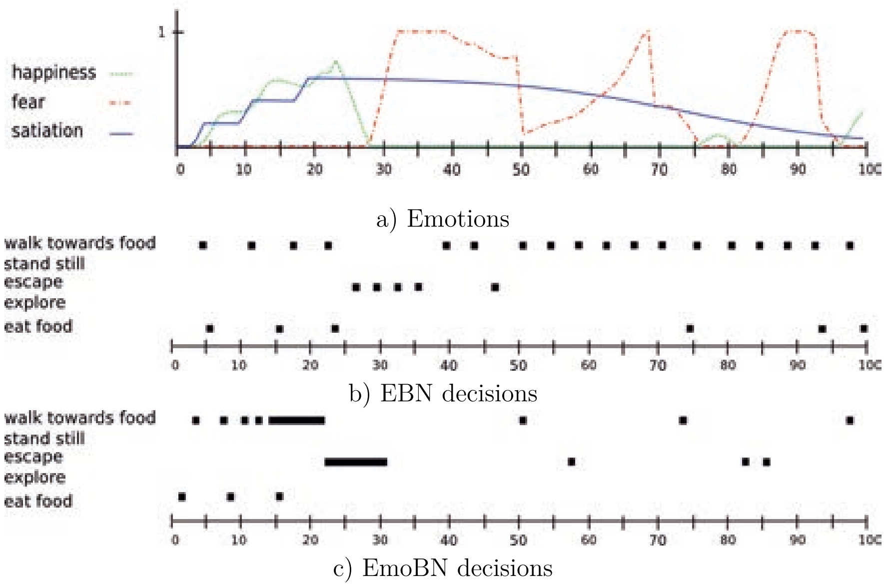
"Why are you putting your shoes on?"
"I have to go to work."
"Why do you have to go to work?"
"To get money."
"Why do you want money?"
"To be able to go on that trip to London next summer."
"Why do you want to do that?"
"Well, it's just fun, I guess."

While many of our choices are not consciously made to improve our mood, subconsciously the prediction of the future consequences for our emotional state is a key factor in decision making. Because of this, in Paper A it is suggested that all the goals in the behavior network should be affective.

Results

In Paper A, two tests are constructed to illustrate the proposed model. In the first scenario, two behavior networks are created, one using an EBN, and one using the proposed EmoBN. Both behavior networks implement similar behavior: the character walks around looking for food and tries to escape if an enemy gets too close. The results of the simulation can be seen in Figure 4.1. In Figure 4.1 a) the emotional states of the character are shown. Figure 4.1 b) and Figure 4.1 c) depict the actions chosen by the character with EBN and with EmoBN, respectively. The most noticeable effect is the difference in the intervals between chosen actions. This has to do with the impact of *fear* on the parameter $\Delta\theta$. For instance, in the interval 30-50, where fear is high, the spacing between the actions increases dramatically.

Note that in this simulation, only one activation spreading cycle is performed per agent update.

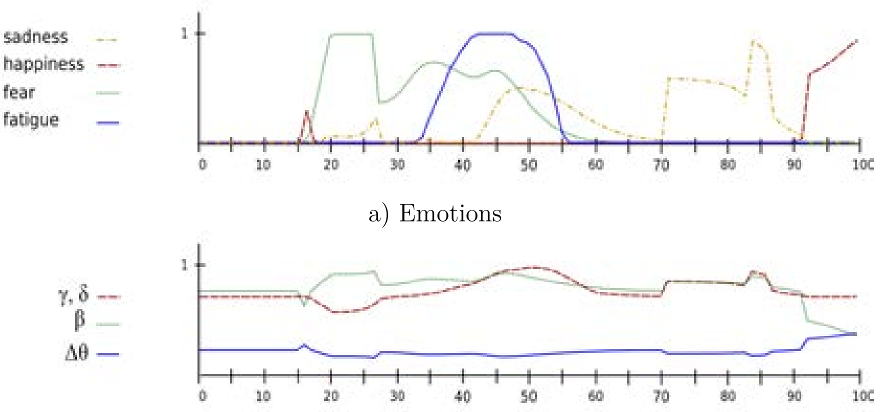


a) Emotions

b) EBN decisions

c) EmoBN decisions

Figure 4.1: Results from scenario 1, from top to bottom: a) the emotional states during the simulation, b) the actions chosen by the EBN character, and c) the actions chosen by the EmoBN character.



a) Emotions

b) Parameters

Figure 4.2: Results from scenario 2, from top to bottom: a) the emotional states during the simulation, b) the values of the parameters.

In the second test scenario the character should attempt to attack the enemy, while at the same time trying to survive. There are health packs that the character can use when the health level gets too low. The purpose of this scenario is to illustrate how the individual parameters change as a result of the emotional impact. The results can be seen in Figure 4.2 b). Figure 4.2 a) shows the corresponding emotional values during the simulation. It can be seen that when fear increases, γ and δ decrease, making planning further ahead more difficult. Furthermore, when happiness increases, β decreases, making the possibility of switching behavior higher.

Comments on the Paper

Note that the emotions used to affect the parameters γ and δ of the network, as mentioned in Table 4.1, incorporate psychological findings of affective influence on time-discounting as well. These emotions are excluded as a result of the work in Paper C (the results of Paper C are discussed in Section 4.2.2).

Summary of Contributions

The work in Paper A extends EBNs to include emotional impact according to the emotional model proposed by Loewenstein and Lerner [LL03]. In short, the following extensions are made. The effect probabilities are affected by the current emotional state of the character. The parameters of the behavior network are influences by emotions. The notion of emotional influences are introduced as a way of affecting behavior modules without affecting the activation of the modules. Moreover, the goals of the behavior network are entirely

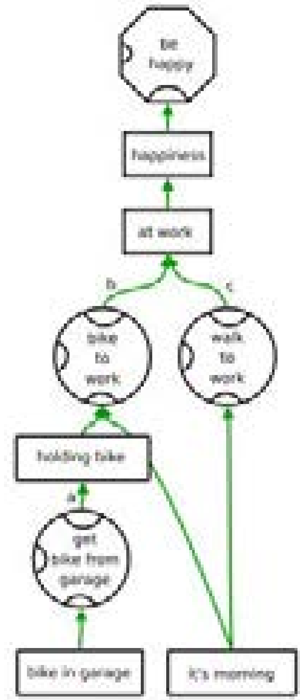


Figure 4.3: A behavior network simulating two ways of getting to work. © 2010 IEEE.

emotional. Parts of the model of Loewenstein and Lerner are implemented, namely the links a, d, e, f, g, h, and i (see Figure 2.6 on page 20).

4.2.2 Time-Extended Emotional Behavior Networks

It is well-known that humans take time into consideration when making a decision. If there is a choice between two actions leading to the same outcome, and one of the actions produces the outcome more quickly, then that action is most likely chosen. It is also clear that humans value effects further into the future less than effects nearer in the future. For instance, people may choose a 50€ reward today over a 100€ reward in two weeks. The tendency to value future rewards less than current rewards is called *time-discounting*. Time-discounting has been studied in economics, neuroscience as well as psychology. It has become apparent that emotions greatly influence the amount of time-discounting a person does (for more information, see Section 2.3.2).

The work in Paper C introduces time-discounting in emotional behavior networks (see Section 4.2.1 of this thesis for an overview of the method). In original behavior networks

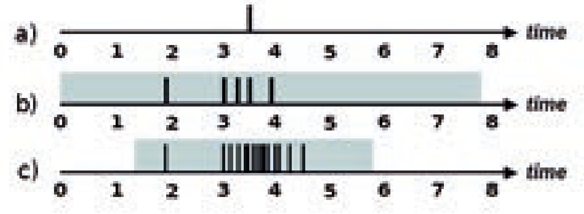


Figure 4.4: Time intervals when having knowledge about a) 1, b) 5, and c) 15 effect delay times. © 2010 IEEE.

(MASM as well as EBN), the effects of behavior modules are assumed to be instantaneous. In practice, however, the effects of a behavior are usually some time in the future. The *effect delay time* should be taken into consideration when a character makes a decision. Given the behavior network in Figure 4.3, an EmoBN as defined in Paper A would always choose walking to work over biking to work (given that the probabilities are all 1, and $\delta < 1$) because it consists of a longer sequence of actions. That biking to work is quicker is not taken into consideration at all.

Effect Delay Time Interval

An extension to EmoBNs where each behavior effect is coupled with an *effect delay time interval* is suggested in Paper C. This is interpreted such that the result of the behavior will likely happen within this time interval after the behavior has been executed. The interval itself is calculated using statistical methods of previous effect delay times. Using the mean value and the standard deviation, a 95% (or more) interval is created. An example of this can be seen in Figure 4.4. In Figure 4.4 a) the character has only one previous outcome to look at, making statistical methods useless. In Figure 4.4 b), however, five different samples for the effect delay time are used, giving a rather broad interval. In Figure 4.4 c) 15 samples are known, narrowing down the interval further.

The effect delay time samples can be gathered by the character itself as time passes by, using some form of evaluation. Such an evaluation would require the character to know when a state has been fulfilled. This is non-trivial as the states consist of fuzzy values. The work behind Paper C originally intended the effect delay time samples to be gathered using a learning module. However, this module failed to achieve the desired results and was later abandoned. Because of this, the effect delay interval can also be manually specified by the behavior designer.

Calculation of Activation

Once the effect delay time interval is known, an estimated effect delay time must be chosen from it. The choosing of this effect delay time is greatly influenced by the emotional state of the character. If the effect is positive from the character's point of view, and the character is in an optimistic mood, the character will choose a shorter time from the interval. In other words, when it is in an optimistic mood, the character believes positive effects will occur more quickly. The opposite is true when the character is in a pessimistic mood. The emotional impact E_I on the decision is calculated as

$$E_I = \frac{1}{n}\sum_{i=1}^{n} pos_i - \frac{1}{m}\sum_{i=1}^{m} neg_i$$

where $pos_1, \ldots, pos_n$ are the emotions that increase optimism and $neg_1, \ldots, neg_m$ are the emotions that decrease optimism. The chosen time t_{emo} from the interval $[I_l, I_u]$ is calculated as:

$$t_{emo} = I_l + \frac{I_u - I_l}{2} \cdot (1 - E_I \cdot sign(benev) \cdot c)$$

where c is a parameter that determines the amount of emotional impact and *benev* is the perceived goodness of the effect. Given the time t_{emo}, the time-discounting factor T_f is calculated. In Paper C the use of hyperbolic time-discounting is proposed, as it has been shown to be more accurate [KS03, KM95]:

$$T_f = \frac{1}{1 + r(1 + E_k)t_{emo}}$$

where r is a factor that is chosen to fit the time scale of the simulation and E_k is the emotional impact on time-discounting calculated thus:

$$E_k = \frac{1}{n}\sum_{i=1}^{n} pos_i - \frac{1}{m}\sum_{i=1}^{m} neg_i \tag{4.1}$$

where $pos_1, \ldots, pos_n$ are the emotions that give a higher time-discounting and $neg_1, \ldots, neg_m$ are the emotions that lower the time-discounting. The time-discounting factor T_f affects the calculation of activation (see original equations on page 13) in the following way:

$$\begin{aligned}
a_{kg}^{t}\prime &= \gamma \cdot I_g \cdot prob_{emo} \cdot T_f \\
a_{kg}^{t}\prime\prime &= -\delta \cdot I_g \cdot prob_{emo} \cdot T_f \\
a_{kg}^{t}\prime\prime\prime &= \gamma \cdot \sigma(a_{jg}^{t-1}) \cdot prob_{emo} \cdot (1 - \tau(p_j, s)) \cdot T_f \\
a_{kg}^{t}\prime\prime\prime\prime &= -\delta \cdot \sigma(a_{jg}^{t-1}) \cdot prob_{emo} \cdot \tau(p_j, s) \cdot T_f
\end{aligned}$$

In general, this means that the activation for a module will decrease if the effect of that module is further away in time.

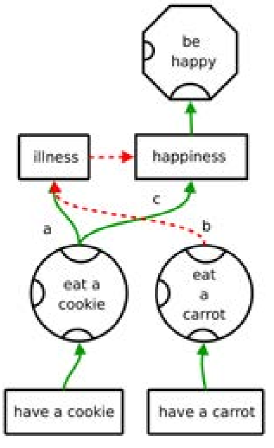


Figure 4.5: A behavior network simulating delayed gratification with emotional impact. © 2010 IEEE.

Case Studies

Two scenarios to illustrate the usefulness of the proposed model are presented in the paper. First, an example is designed to show the ability to reason about time when making decisions. The behavior network used can be seen in Figure 4.3. The times for the links a, b, and c are given, with $a + b < c$. The example demonstrates that the agent is now able to take time into consideration when choosing between actions, correctly choosing to take the bike to work rather than to walk.

The second scenario illustrates the emotional impact on time-discounting. The behavior network used is depicted in Figure 4.5. The example demonstrates an important psychological and emotional effect related to delayed gratification. Eating unhealthy food now may lead to negative health effects in the future and should therefore be avoided from a rational point of view. However, eating unhealthy food will also give instant gratification. The balance between taking future consequences into account and giving in to momentary satisfaction is greatly influenced by the emotional state of a person. In this example, by using the proposed time-extended behavior network, the character chooses to eat a cookie when it is sad, but eats a carrot when it is happy. This is consistent with psychological theories concerning over-eating [SAGA00, Gan89, PT96, FAS04].

Corrections to Paper

In Paper C there is an error regarding the emotions used to affect time-discounting. Sadness is incorrectly mentioned as an emotion that produces *less* time-discounting, when it actually gives more time-discounting, i.e. makes a person prefer immediate rewards over delayed rewards. The results in the paper nevertheless correctly displayed the anticipated results. This was due to a logical error in another part of the behavior network module. When the error was properly addressed, the simulation yielded anticipated results.

The logical error is connected to the use of “virtual behavior modules” for implications. Behavior modules are fundamentally different from implications because a behavior does not have to be selected even though its preconditions are true. An implication, however, is always implicitly executed. $A \rightarrow B$ implies that if A is true, then B is also true. Because of this, the activation spreading does not apply to implications. Other formulas must be used instead.

Summary of Contributions

In Paper C, EmoBNs are extended to include time-discounting with emotional influence using psychological theories. The extensions enable the character to reason about time when making a decision. Furthermore emotions affect the amount of time-discounting that a character does. The usefulness of the extension is supported by giving two examples of situations the proposed extension can model, which was previously not possible to model using EmoBNs.

4.2.3 Comparison of Emotional Behavior Networks and Behavior Trees

EBNs are extended in Paper A and Paper C to include emotional and temporal impact. Behavior networks, however, are not widely used within the gaming industry. Instead, behavior trees have been used extensively as an action selection mechanism within the computer game industry in recent years. It is the aim of Paper F to compare the two models, EmoBNs and behavior trees, in terms of functionality and management.

Comparison in Functionality

Paper F begins with a comparison of the two models in terms of functionality.

Planning Planning is inherent in EmoBNs. Although the planning is implicit, a chosen action is always linked to one or more goals through a series of other actions. The notion of preconditions and effects allows the virtual character to reason about the best possible action to take at that particular moment, rather than to follow hand-coded rules.

Behavior trees often perform the actions in sequences. These can be thought of as plans, but they are plans hand-coded by the game designer. These plans cannot be remade on the fly should changes in the environment occur. To allow for all possible interruptions to the plan, one must create a large behavior tree, which is difficult to both design and manage.

Goal Awareness In EmoBNs, the virtual character is aware of what goals it is pursuing. While the top goals may be too general to be useful, it is possible to deduce which subgoals influence the currently selected behavior.

Behavior trees have no explicit notion of goals. It is possible to view each interior node of the tree as a subgoal that must be satisfied, but this is of little use to the virtual character's decision making and affective appraisal, as they cannot be easily mapped to a world state.

Notion of Time In EmoBNs, time is taken into consideration when making a choice. A sequence of actions taking a shorter time is automatically chosen over a sequence of actions that take a longer time to perform (should all other properties be identical). While behavior trees can be manually designed so that faster actions precede more time-consuming actions, behavior trees handle dynamic changes to the environment poorly. Assume two sequences of actions, $\langle A, B, C\rangle$ and $\langle D, E, F\rangle$, take the times 2, 3, 3 and 5, 3, 2 time units respectively to complete. Both sequences lead to the desired result. The second sequence takes a longer time to complete, probably leading to a design of the behavior tree that chooses this sequence as a second alternative. However, suppose that the result of D is already achieved due to incidental environmental factors. The sequence $\langle E, F\rangle$ is shorter than $\langle A, B, C\rangle$ and should be chosen instead. The EmoBN can handle this, but the behavior tree cannot.

Fuzzy Knowledge Fuzzy knowledge is useful in many situations. Many concepts cannot easily fit into the binary true/false category. Age, social status, distance, and wealth are a few of the many concepts that work well to classify using fuzzy logic.

EmoBNs incorporate fuzzy knowledge automatically, since all states can be represented as continuous values in the interval $[0, 1]$. Behavior trees on the other hand cannot easily take advantage of fuzzy values. Fuzzy values can be used in the calculations within a condition in the behavior tree, but must produce a boolean value in the end, since a condition in a behavior tree must be expressed as a boolean.

Emotional Decisions Emotions greatly affect human decision making. Section 2.3.2 summarizes the theory concerning affective decision making in humans. Furthermore, having emotional characters in a game enhances the believability of the players as well as the game immersion.

EmoBNs incorporate emotions into the decision making in a direct as well as an indirect manner, following the work in Section 2.3.2. In behavior trees, on the other hand, there is no straightforward way to introduce emotions except as simple conditions.

Unpredictable Behavior One common complaint concerning NPCs is their predictable behavior. Unpredictability is desirable from a gameplay point of view, but it is important that the actions can be explained from the player's point of view.

EmoBNs are more unpredictable due to their emotional impact and the dynamic and continuous nature of the model. Behavior trees are by nature static and simple randomness is often introduced to break predictability. However, behavior trees do not support more sophisticated unpredictability.

Design and Management

Paper F continues with a comparison of the two models with respect to design and management.

Behavior Design Behavior trees are easy to use and the results are predictable. The simplicity of the technique also makes them easy to debug. EmoBNs on the other hand can be difficult to manage. Achieving the desired result can be difficult due to the numerous parameters that must be tuned.

Scalability and Computation Times Behavior trees scale well in terms of computation. Due to the tree structure it is also rather easy to maintain an overview of the behavior of the NPC even as the size of the tree grows. EmoBNs scale up fairly well in terms of computation, having a linear scaling most of the time. However, managing the network as the size grows is difficult and the tuning of the parameters can become burdensome.

Discussion

Paper F concludes by summarizing the difference between behavior trees and EmoBNs. It is suggested in the paper to use behavior trees when a large number of actions are used and full control is required, and EmoBNs when the need for dynamic and interesting behavior is prominent.

Summary of Contributions

The work in Paper F compares behavior trees with EmoBNs in terms of functionality and management. The two models both have benefits and drawbacks, signifying the need for new techniques that merge the benefits from both models, while limiting the drawbacks.

4.2.4 Emotional Behavior Trees

The results of the comparison in Paper F suggest that behavior trees contain many components that are useful for a game designer. They are easy to use, execute fast, and scale up well. However, one of the drawbacks of behavior trees is that they are fairly static by nature. It has not been suggested how to include an emotional component in behavior trees, except as trivial conditions (e.g. $happiness > 0.5$). The aim of Paper G is to address this problem by introducing an emotional selector in the behavior tree. This enables the game designer to choose which part of the behavior tree that should be under emotional influence. The rest of the behavior tree remains the same. The proposed extension is named emotional behavior trees (EmoBTs).

Emotions influence decision making in many ways. It is suggested in Paper G to use three specific factors that are relevant to emotional decision making in games: planning, risk, and time. Emotions affect how humans view these aspects when selecting the course of action (for further information, see Section 2.3.2). For instance, happy and angry people are willing to accept greater risks than people who are sad. It is also known that happy people do less time-discounting than sad people, which means that happy people tend to go for options that offer future rewards whereas sad people choose options that give quick rewards. When planning, sad people focus more on bottom-up processing, leading to a slower and more cumbersome planning.

Emotional Selector

As an extension to behavior trees (as described in Section 2.1.3), the work in Paper G introduces a new type of selector, the *emotional selector*. This selector orders its child nodes with respect to the three factors: the inherent risk, the time it will take to achieve the result, and the planning effort involved. The emotional selector works in a similar way to the regular priority selector, except that the priority is ordered according to the three factors mentioned above. The ordering is done at run-time, enabling the impact of the character's current emotions on the decision making.

Planning effort

Behavior trees involve no actual planning process. However, each sequence of actions can be thought of as a constructed plan. It is suggested in Paper G that this is can be used to simulate the planning effort *as if* actual planning had been performed. Each leaf node of the tree must be given a *planning value* by the game designer. The planning value for the internal nodes are derived in the following way.

Action An action has a planning value that is set by the designer (1 by default). Some actions may require more planning than others and it is vital that the game designer can choose an appropriate planning value for each individual action node.

Condition A condition has a planning value that is set by the designer (0 by default).

Sequence Selector A sequence selector is designed to perform all its children in sequence. This can be thought of as a plan. The planning value is defined as the sum of the planning values of its child nodes.

Priority Selector It is impossible to know which of the child nodes a priority selector will perform, hence the planning value of the priority selector is defined as the average value of its child nodes' planning values.

Parallel Node A parallel node executes, or tries to execute, all of its child nodes in parallel. This means that the plans for these child nodes must have been established prior to execution. Therefore, the planning value of a parallel node is defined as the sum of the planning values of its child nodes.

Decorator The planning value of the decorator node is the same as the planning value of its child node.

Risk Assessment

The risk value assigned to each node corresponds to how dangerous the virtual character thinks performing this action is. A risk value has a value between 0 and 1; 0 being no risk at all, and 1 being highly dangerous. The risk value roughly corresponds to the probability that something bad will happen. Behavior trees themselves cannot know how dangerous an action is, so leaf nodes of the tree must be assigned a risk value by the game designer. The risk values of the rest of the nodes are derived from the leaf nodes in the following way.

Action An action has a risk value that is set by the designer (0 by default).

Condition A condition has a risk value that is set by the designer (0 by default).

Sequence Selector A sequence selector ideally executes every child node in the sequence, hence the overall risk is the total risk. This is calculated as $1-p$ where p is the probability of nothing bad happening.

Priority Selector A priority selector only successfully executes one of its child nodes. It is impossible to determine beforehand which node will be executed, hence the risk value of the priority selector is defined as the average of the risk of the child nodes.

Parallel Node All of the child nodes of a parallel node are executed, hence the risk is defined as the combined risk of all the nodes.

Decorator The risk value of a decorator is the same as the one of its child node.

Time

Each node in an EmoBT has a time interval representing the time interval the action will be completed within. As with the risk value, the time interval must be given by the game designer for the leaf nodes.

Action An action has a time interval that is set by the designer ([0 0] by default).

Condition A condition has a time interval of [0 0] because conditions are always instantaneous.

Sequence Selector The time interval of a sequence selector is the sum of the lower and upper limits of the time intervals of all the child nodes.

Priority Selector Since one cannot determine which child a priority selector will ultimately successfully execute, the largest possible time interval is used.

Parallel Node All child nodes of a parallel node are executed simultaneously, hence the time interval is the maximum of its child nodes.

Decorator The time interval of a decorator is the same as the time interval of its child node[2].

Adding Emotional Impact

Given the three different factors, the EmoBT model calculates four corresponding emotional impacts. These variables are computed using appropriate emotions that affect the three different factors, and emotions that influence optimism. Each emotional impact is calculated as the average value of the emotions that affect the factor positively minus the average value of the emotions that affect the factor negatively. Three different weights are calculated, one for each factor. The weight for risk is calculated as:

$$W_{risk} = (1 - E_{risk} \cdot \delta) \cdot risk$$

where $risk$ is the risk factor for the node, E_{risk} is the emotional impact on risk, and δ is a constant that affects how much emotional impact there should be. The weight for time is calculated as:

$$W_{time} = (1 - \frac{1}{1 + \mu \cdot time}) \cdot max((1 - \lambda + \lambda \cdot E_{time}), 0)$$

where μ is a parameter whose value depends on the time scope of the simulation. The first part of the formula represents a hyperbolic time-discounting function. The variable *time* is the emotional effect delay time calculated from the node's time interval $[L, U]$:

$$time = L + \frac{U - L}{2}(1 - \sigma \cdot E_{opt})$$

where E_{opt} is the emotional impact on optimism. The weight for planning is calculated as:

[2]Ideally, one would be able to set some form of calculation formula as a game designer. For instance, a 4-times loop decorator should have its child node's time interval multiplied by four. In other examples, the decorator may provide an even more complex behavior.

$$W_{plan} = (1 - \frac{1}{1 + \omega \cdot plan}) \cdot max((1 - \phi + \phi \cdot E_{plan}), 0)$$

where E_{plan} is the emotional impact on planning. In the equations above, λ, δ, σ and ω are constants that affect how much emotional impact there is.

The tree weights W_{risk}, W_{time}, and W_{plan} are combined to create a total weight for each node. The weights are constructed in such a way that a smaller weight is a more desirable option. Therefore, all child nodes of an emotional selector are ordered in ascending order with respect to the combined weight. There are only three different factors that distinguish the child nodes from each other. However, an emotional selector can contain many more child nodes. Because of this, it is suggested in Paper G to use a randomized component in selecting which child node to execute. Each child node is assigned a probability according to its place i in the weight-ordered list:

$$prob_i = a(1 - a)^{i-1} \quad \text{with } 0.5 \leq a < 1$$

where a is chosen depending on which distribution one desires. For instance, using $a = 0.5$ results in the probabilities $0.5, 0.25, 0.125, \ldots$ for the nodes. The last node receives the remaining probability, to ensure a 100% coverage. Using the assigned probabilities, the child nodes of the emotional selector is randomized into a fixed order. After this ordering, the emotional selector behaves as a regular priority selector. When the emotional selector has completed its execution and is executed again, the emotional impacts and the ordering are recalculated.

Results

An example using a virtual character for a fighting scenario is presented in Paper G. A fighting NPC often has several different ways to attack an enemy, each option involving different amounts of risk, planning and time. It is therefore a good example to use to illustrate the usefulness of the proposed model. In the example in Paper G the NPC can choose the following attacks. It can throw grenades, use a musket, use a sword, or do a fancy knifing maneuver. The emotional behavior tree used for the example is depicted in Figure 4.6. The risk, planning, and time interval values for the respective actions are displayed in Figure 4.7.

The fighting example above is simulated under different emotional states. The results are listed in Figure 4.8. In Figure 4.8 a), the weight values for each action are shown under different emotional conditions. It can be seen that the weight values change widely due to emotional impact. For example, when the character is afraid, maneuvering is a good choice because it is not risky. When the character is sad and tired, throwing grenades seems like a good option because it does not involve much planning and it gives fast results. In 4.8 b) the generated probabilities for each child node are displayed, using $a = 0.7$. Note that the ordering with respect to weight value is preserved, but there is a greater distinction in which action is more preferable.

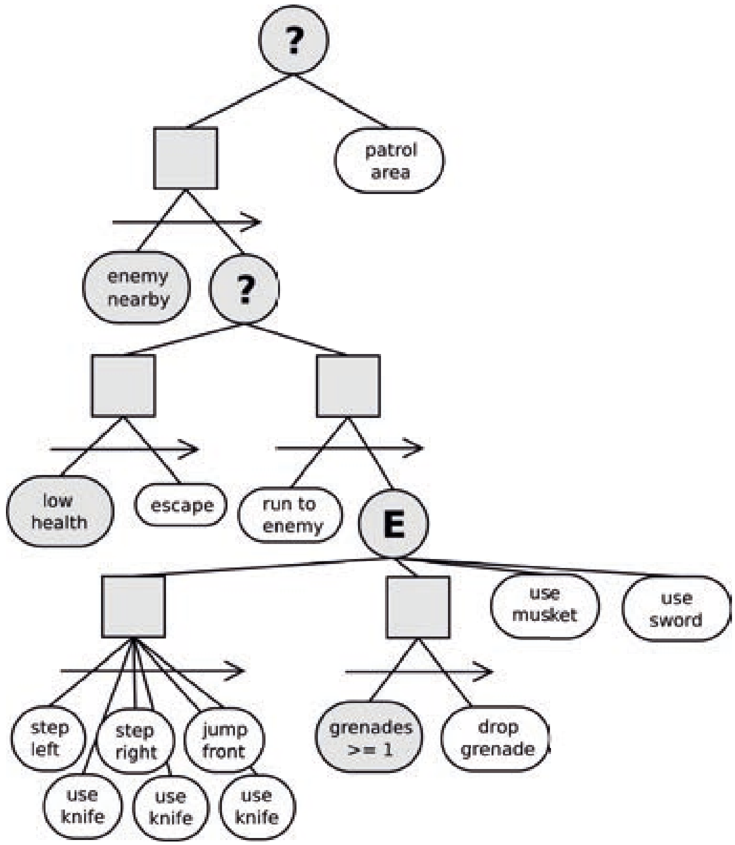
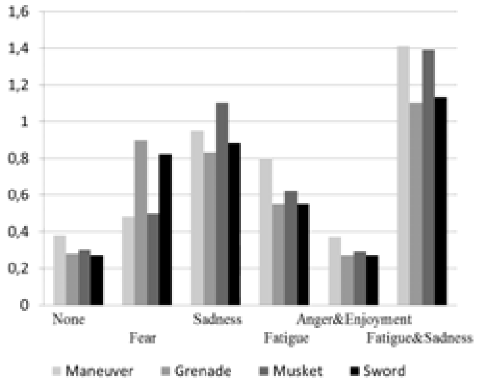


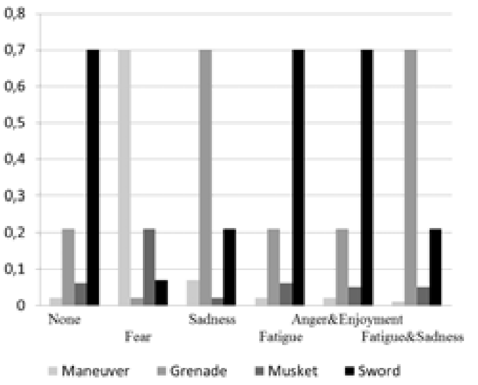
Figure 4.6: The emotional behavior tree for the fighting example. © 2012 IEEE.

	use knife	*Movement Actions*	grenades >= 1	throw grenade	use musket	use sword
Time	**[0, 0]**	**[0.33, 1.0]**	**[0, 0]**	**[1.5, 2.5]**	**[7, 10]**	**[2, 3]**
Risk	**0.033**	**0.0**	**0.0**	**0.7**	**0.2**	**0.6**
Plan	**1**	**1**	**0**	**1**	**1.7**	**1**

Figure 4.7: Time, risk and planning values of leaf nodes. Movements actions refer to step left, step right, and jump front actions. © 2012 IEEE.



a) Weights



b) Probabilities

Figure 4.8: a) The combined weight values for the fighting game example, given different emotional states. When listed, each emotion has the maximum value 1. b) Resulting probabilities of the fighting game example with $a = 0.7$. © 2012 IEEE.

Summary of Contributions

Introducing emotions into behavior trees as suggested in Paper G enables a more dynamic and changing behavior. By taking advantage of known psychology theories concerning emotions, the EmoBT lets emotions affect the decision making in a human-like and intuitive way.

The proposed emotional selector is meant to be used when the child nodes are of similar type (such as different way to attack an enemy). Using a purely random selection of available actions is a more common method used in games, but is not as useful, as simple randomness provides little believability for the player. The benefit of the proposed model is that only parts of the behavior tree, the parts the game designer has chosen, are influenced by the emotions. This enables the game designer to maintain full control over the rest of the behavior tree, leaving only a subpart open to emotional decision making.

4.2.5 Related Work

While emotions are widely used in architectures for virtual characters, it is rare that they are incorporated into the decision-making mechanism. Frequently, emotions are used to guide the animation of the character [BAZB02, EKMT03] or the emotional expressions of the robot [LM07].

Dias and Paiva [DP05] suggest a model for virtual characters where emotions play a large role in the coping mechanism. Since the agent copes with changes to its plans and intentions by triggering and using emotions, the overall behavior of the character is influenced by emotions. The proposed model does not appear, however, to incorporate emotions directly in the planning of the agent. In later work by Aylett, Dias and Paiva, emotions are tied to the planning process [ADP06]. Emotions are triggered by the planning process itself but the emotions also affect which plans are chosen for execution. In general, intentions that trigger the strongest emotions are the ones the character should choose to attend to. Therefore the character selects a plan that corresponds to the most important intention. In their system there is also a reactive layer. This layer consists of a set of action rules which can be triggered by specific emotions. The strongest emotion will precede over other emotions. In EmoBNs, by contrast, emotions affect the entire planning process, from emotional probabilities, time-discounting, and emotional goals. Emotions are fully integrated into the planning process in a way that mimics human decision making. The EmoBNs are not involved in the process of triggering emotions, however. EmoBTs do not perform planning, but emotions are involved in how the character chooses between different courses of action. Similarly to the method proposed by Aylett, Dias and Paiva, the EmoBT lets emotions decide which alternative to choose. However, it lets all relevant emotions affect the decision making instead of focusing on only the strongest emotion.

Marsella and Gratch suggest a model where emotions influence the coping of the virtual character [MG02]. The influence of emotions on decision-making is through coping strategies, such as “shift responsibility” when the character feels it has performed poorly at a task. Emotions appear to have no broader impact on the decision making in their model. In their earlier work on creating believable virtual characters [GM01], the focus lies on the appraisal process and how the resulting emotions affect the animation of the characters. The coping behavior can be compared to the emotional goals in the EmoBN. In EmoBNs, the agent tries to minimize unpleasant emotions and maximize good emotions. Coping works in a similar, although more specific, way. EmoBTs do not model coping. Instead, they focus on the selection between alternatives leading to the same result, letting the emotions affect which one to choose.

Henninger et al. [HJC03] propose a model where emotions are incorporated into the decision-making process. The appraisal process results in emotions which are mapped onto two dimensions: confusion/clarity and pain/pleasure. Ultimately the emotions are mapped into a single dimension, representing the level of arousal. Arousal is the only emotional impact in the system, with a large value of arousal narrowing down the agent’s focus of attention. In contrast, EmoBNs let a variety of emotions affect the decision making in numerous ways, taking inspiration from psychology theories. EmoBTs also utilize several emotions in the decision making, using ideas from psychology on how emotions affect the perception of risk, time, and planning.

Velásquez [Vel97] suggests a computation model for emotions, where the main focus is on the trigger of, and interplay between, emotions. The suggested decision-making model is rather vague and focuses mainly on actions that are a direct result of the emotions currently felt (e.g. the agent wants to hit someone because it is angry). In contrast, EmoBNs fully incorporate emotions into the decision making of the character. EmoBTs also use emotions in a more complex way, letting several emotions automatically affect the character’s decision.

Yu and Choi [YC05] propose a method where emotions and personality affect decision-making for robots. They focus on the interplay between a human being and the robot, using rewards and punishments to affect the emotions of the robot and in turn change its personality in the long run. The personality of the robot affects what emotions are triggered and also how these are displayed to the environment. It is unclear how they let emotions affect the decision making as the range of actions available to the robot only consists of actions used to display emotions (e.g. the robot wagging its tail or shaking its head). It appears that the actions are chosen with respect to an extrovert personality trait of the character, which in turn changes according to the emotional state of the robot, coupled with the strongest emotion currently felt by the robot. EmoBNs and EmoBTs both let emotions affect the decision making without first affecting the personality of the character. In other respects, it is difficult to compare EmoBNs and EmoBTs to the approach by Yu and Choi as the types of actions available are only used to display emotions outwardly.

4.3 Pathfinding

This section covers the third contribution of this thesis, i.e. covert pathfinding and emotional pathfinding. Covert pathfinding concerns the ability for a virtual character to move through a world as stealthily as possible, using knowledge about previous enemy positions. The emotional pathfinding involves emotional memories as well as the current emotional state as guides for the character when moving through the world.

The work in this section is presented in detail in Papers D and E.

4.3.1 Covert Pathfinding

Pathfinding is essential for most virtual characters. Most of the time pathfinding tends to focus on finding the shortest path from point A to B. This, however, is hardly natural or human-like. While humans tend to want to shorten their paths to a certain extent, many other factors have an impact on what routes a person chooses through the environment. One of the things that are of great importance, especially in hostile environments (which may not be so common in an ordinary person's life, but are all the more common in computer games) is moving from point A to point B while at the same time trying to stay out of sight.

In Paper D covert pathfinding is performed with enemy characters in mind. The algorithm is applicable to other scenarios, however. For instance, the virtual character may want to avoid being detected by bullies, noisy landlords or the police. In the following sections, where the algorithm is described, the word enemy can mean anything that the virtual character wants to avoid being seen by.

General Visibility Maps

Much work has been done on creating general visibility maps, i.e. maps that can be used to determine how much a point on the map is visible to other points on the map. Isovists

Figure 4.9: A simple isovist. The black boxes are obstacles and the black dot is the point from which the isovist is calculated.

[Ben79] (a kind of visibility map) are used to calculate the visibility from one particular point. Often the results from many isovist calculations are merged into one visibility map to show the overall visibility of an environment. Isovist calculations have been used extensively within architecture design to plan structures with desired visibility. An example of a simple isovist can be seen in Figure 4.9.

When creating a visibility map with isovists there are many different computational methods one can use. For instance, one can choose the average distance to the nearest obstacle in all directions. One can also choose the shortest distance, the largest distance and so on. A good review of the different computations used and the results they yield can be found in [Bat01]. Furthermore, van Bilsen [vBS08] and Stolk show that the average error when using a discretized version of the world compared to a polygonal representation is only roughly 11%. This error is too large for visibility calculations for building projects where the calculations need to be precise, but for the purpose of this work the error is acceptable.

An example of a general visibility map can be seen in Figure 4.10. This example shows the layout of a simple house. As can be seen, the crossings of hallways provide good visibility, while narrow corridors reduce visibility quite heavily.

Probability Map

In Paper D, knowledge about previous enemy positions is used to create a visibility map. This knowledge can be gathered by one character or it can be knowledge shared between characters on the same "team". What is interesting is the general whereabouts of the enemies, not the individual positions. A map that shows the probability distribution of the enemies' positions is calculated using previous known positions of enemies. A small falloff radius is used around each position to make a smoother map and to mimic the way humans remember things (humans do not remember positions exactly, but rather focus

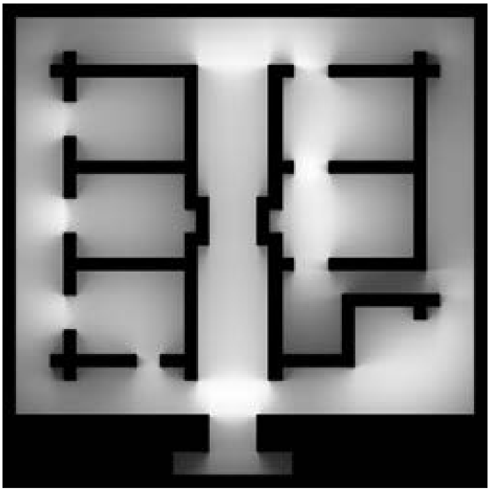


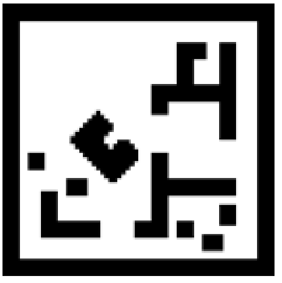
Figure 4.10: An example of a general visibility map. The lighter a point on the map is, the more that point is visible from the rest of the map.

on the general whereabouts of a person or an object). The calculation of the probability map scales linearly to the number of known enemy positions and can therefore be computationally heavy if the entire map needs to be recalculated often. However, it is possible to incrementally add currently known enemy positions to the visibility map, avoiding the recalculation of the entire map. This also removes the necessity of a long-term memory for the characters, enabling a more architecture-independent approach. An example of a probability map can be seen in Figure 4.11 b).

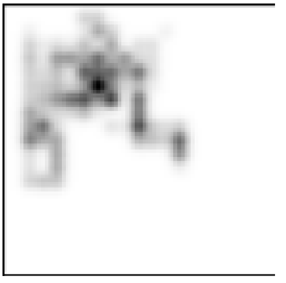
Knowledge-Based Visibility Map

In Paper D, the probability map is used to calculate a visibility map. A value on the covert visibility map signifies how visible this point is from positions enemies are known to have. Using all the nodes in the probability map to calculate visibility is rather cumbersome, however, so only a subset of the nodes is used. Nodes with higher probability are preferred over nodes with lower probability. The algorithm for creating the visibility map given the probability distribution of enemy positions is the following. For each point in the probability map that exceeds a threshold value h, a discrete isovist is calculated and added to the visibility map, weighted by the probability value of that particular point. Finally, the visibility map is normalized. The variable h will determine how many nodes are taken into account when calculating the visibility map. Using a higher value of h will lead to a faster, but less accurate, visibility calculation.

In Paper D an example scenario is presented. An environment as the one shown in Figure 4.11 a) is created, and the enemy probability map used can be seen in Figure 4.11



a) Obstacles



b) Probability map

Figure 4.11: a) The obstacle map of the environment. b) A 64 by 64 probability map calculated from 410 enemy positions. A darker color means a higher concentration of enemy positions.

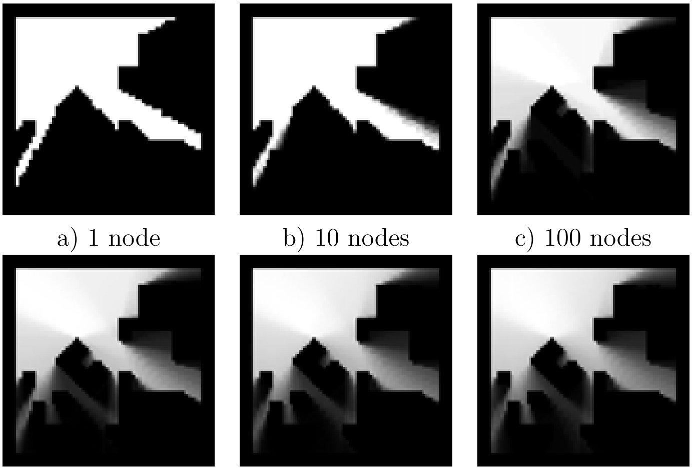
b). The resulting visibility maps, given different values of h, are shown in Figure 4.12. In this case, using 300 nodes will give a result that is nearly indistinguishable from using all nodes. Using only 300 nodes instead of the full 4096 nodes speeds up the calculation of the visibility map by a factor of 11.

Next, the use of a falloff function for the distance between two points when calculating the visibility is suggested in Paper D. Instead of letting a point 10 units away contribute the same amount to the visibility of the current point as a point 2 units away, it should contribute less. A linear falloff function is used to accomplish this. Using a falloff function furthermore speeds up the computation of the visibility map as points further away than a given distance threshold are not taken into account. This is especially useful if the environment the visibility is calculated for is very large, for instance a town. The results of using a falloff function can be seen in Figure 4.13. In Figure 4.13 a) only one node is used to calculate the visibility map. The falloff function is clearly visible. In Figure 4.13 b) 300 nodes are used.

Integration into Pathfinding

For the pathfinding, the work presented in Paper D suggests a hierarchical approach based on a regular grid structure. A*-search is used to find the best path. The cost of the links between the nodes is defined in terms of both distance and visibility. The impact of visibility is parametrized and should preferably be dependent on the emotional state of the character as well as current intentions.

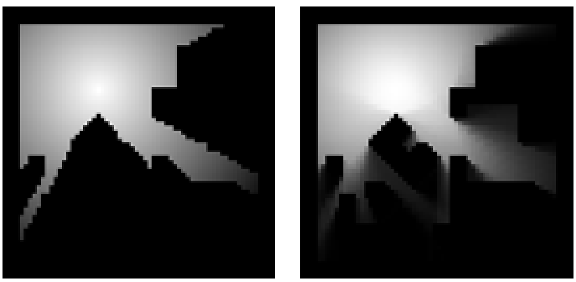
The results of the pathfinding using the visibility map can be seen in Figure 4.14. In Figure 4.14 a), the red dotted path is the path the character would take should the visibility map not be taken into account. The cyan path is the path the character takes when trying to avoid being seen by enemies. It is clearly evident that the character tries to move in areas that are usually less visible to enemies. In Figure 4.14 b) the same path is shown, but with the speed of the characters visible. In the areas where it is potentially



a) 1 node b) 10 nodes c) 100 nodes

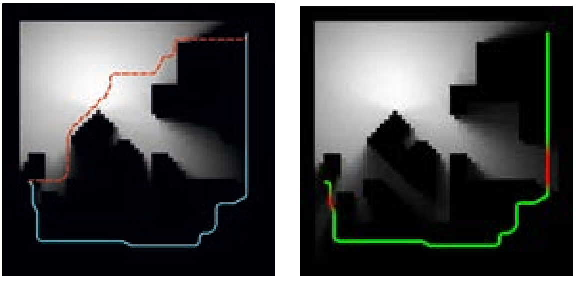
d) 300 nodes e) 500 nodes f) All nodes

Figure 4.12: Visibility maps (size 64 by 64) calculated with different values for the threshold h. Left to right, top to bottom: a) 1 node and $h \approx 1.0$, b) 10 nodes and $h \approx 0.7$, c) 100 nodes and $h \approx 0.5$, d) 300 nodes and $h \approx 0.3$, e) 500 nodes and $h \approx 0.2$, f) all 4086 nodes



a) 1 node b) 300 nodes

Figure 4.13: a) Visibility maps (64 by 64) calculated with respect to distance. Left to right: a) 1 node, b) 300 nodes.



a) Comparison b) Speed

Figure 4.14: a) A comparison between using the Euclidean distance as cost function (red dotted path) and using the visibility cost function (cyan path), b) Character speed (red signifies a faster speed) varies with visibility value.

more dangerous, the character speeds and crouches up to avoid being seen.

Corrections to Paper

The probability maps seen in Paper D as well as in this thesis have a discretized look. This is not an error due to the algorithm presented in the paper, but is the result of a conversion error in another part of the agent architecture. This leads to discretized positions for the enemies.

Summary of Contributions

Covert pathfinding is useful both for believability and for better performance in computer games. If an enemy character in a game tries to hide when it is sneaking closer to the enemy camp, it greatly increases the believability. By using knowledge about previous enemy positions the stealth increases even more. Hiding potentially increases the performance of the characters as they are less likely to be seen and therefore less likely to be attacked. Using only a general visibility map for covert pathfinding is not enough as it may result in strange behavior (where a character tries to hide in a supposedly narrow alley where enemies always patrol). While it is possible to hand-code how to avoid this behavior in games, the approach in Paper D solves this problem in an automatic way.

The use of covert pathfinding is not restricted to typical fighting games. One can imagine potential uses in many different types of games and simulations. A predator sneaking up on its prey (wants to avoid being seen until the last moment), a celebrity sneaking past its fans, a professor trying to avoid being seen by students while exiting a building - the types of situations where covert pathfinding is useful are many and varying in nature.

Using a probability map enables the removal of less influential nodes from the visibility calculation, speeding up the computations considerably. Using a falloff function also speeds up the visibility calculations, particularly in large environments.

4.3.2 Emotional Pathfinding

The work in Paper E extends pathfinding to use emotional memories tied to different geographical locations. Emotions humans feel at the time an event occurs are attached to the memories of that event [LeD96]. This knowledge is exploited to create the concept of emotion maps that in turn affect the pathfinding.

The work in Paper E is inspired by the notion of *influence maps* [MF09, Toz01]. There is no formal definition of influence maps, but in Paper E they are defined as a technique using 2D environmental maps that to assist the characters' decisions. Influence maps have been used extensively in strategy games to make strategic high-level decisions (such as where to place a particular building).

Emotion Maps

The concept of emotion maps is presented in Paper E. A map is created for each emotion that is relevant for the pathfinding[3]. At each instance the agent's current position and its emotional value is used to add information to the map. An S-shaped filter (a Sigmoid filter) is used to first filter the emotional value to avoid small values building up over time. A falloff filter (also a Sigmoid filter) is used to splat the filtered emotional value over an area around the agent's current position.

The different emotion maps must be combined together into one map to be of use in pathfinding. The work in Paper E takes advantage of the knowledge concerning emotional memory retrieval to combine the emotion maps. When in a particular emotional state, it is easier to retrieve memories that were experienced during a similar emotional state [Bow81]. It is also reasonable to assume that certain emotions should have more impact on the pathfinding than others. Fear, for instance, tends to have a high impact on our choices. Therefore some weights need to be introduced. The procedure to calculate the combined emotion map is the following.

- Normalize the emotion maps to the range $[0, 1]$.
- Calculate the combined emotion map out of a set E of emotions maps as follows:

$$SUM = \sum_{i \in E} i \cdot w_i \cdot (1 - a + a \cdot emo_i) \tag{4.2}$$

  where w_i is the weight for the emotion map i. The constant a is used to determine to what extent the current emotions should affect the merging of the emotion maps. emo_i is the emotional value for the emotion represented by the emotion map i at the

[3]Which emotions are relevant is decided by the game designer, and depend on the requirements of the game, the emotion module used, and so on.

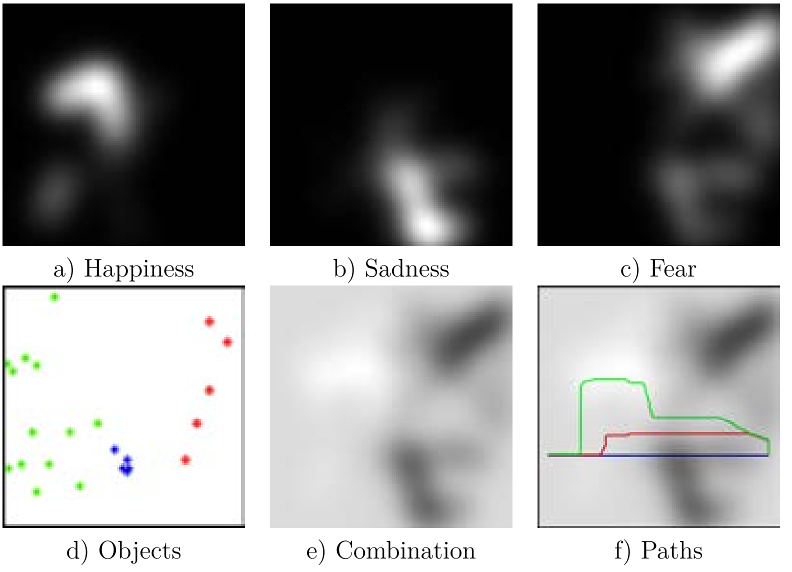


Figure 4.15: From left to right, top down: a) emotion map for happiness, b) emotion map for sadness, c) emotion map for fear, d) objects colored according to emotional attachment e) the combined emotion map, f) resulting paths using different amount of emotional impact.

time the combined emotion map is calculated. The weights can be negative as well as positive. Fear, for instance, should have a large negative weight, implying that fear has a great impact on pathfinding but should make the character *avoid* a place rather than seek it out.

- The resulting map is again normalized to $[0, 1]$.

Results

An example scenario where emotions maps are used is presented in Paper E. The character explores a world filled with objects that trigger different types of emotions. The positions of the objects can be seen in Figure 4.15 d). The green, blue and red dots represent objects that trigger the emotions happiness, sadness and fear, respectively. The resulting emotion maps are depicted in Figure 4.15 a), b), and c). In the figures, the lighter the color, the stronger the emotional value linked to that place on the map. A combined emotion map is shown in Figure 4.15 e). Using the emotion maps in the pathfinding is straighforward. The A*-algorithm uses the combined emotion map to calculate the cost between the two nodes. The amount of influence the emotion map should have on the pathfinding is configurable.

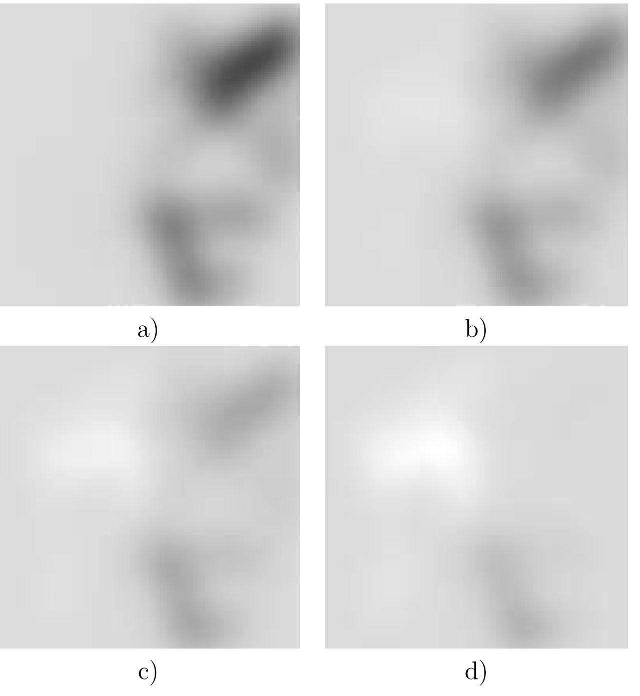


Figure 4.16: Combined emotion maps calculated using full emotional impact and different emotional values. From left to right, top down: a) $fear = 1, happiness = 0$, b) $fear = 0.66, happiness = 0.33$, c) $fear = 0.33, happiness = 0.66$, d) $fear = 0, happiness = 1$. Sadness has a constant value of 0.5 in all maps.

Resulting paths can be seen in Figure 4.15 f). The blue path signifies no use of emotion maps, the red path signifies using the emotion map somewhat, and the green path signifies using emotion maps to a great extent.

In Figure 4.16 four combined emotion maps are displayed, using only different values for the current emotional state of the character. It is easy to see that depending on the current state of the character, the different emotion maps are used to different extents. The impact on the pathfinding will of course be quite profound.

Summary of Contributions

Using emotion maps improves the realism of the pathfinding, especially in games or game-like simulations where the characters have rich personalities and history. These types of characters can be found in role-playing games, such as Oblivion. Seeing how a character prefers to walk through areas where it has experienced enjoyment is both natural and intuitive. Using emotions to affect the decision making in this way furthermore has support in psychology theories concerning emotions. The work in Paper E also uses the current emotional state of the character to blend the separate emotion maps together. This in turn makes the character act on its feelings, avoiding fearful places when it is afraid, and so on.

4.3.3 Related Work

Geraerts and Schager propose a stealth-based pathfinding method [GS10] based on their previous work the Corridor Map Method (CMM) [GO07, GS10, OKG08]. They use the knowledge of where enemies are currently located and the directions they are looking in to create a visibility map. The visibility information is later used in the CMM algorithm. Like the covert pathfinding method proposed in this thesis, they use a regular discrete two-dimensional representation of the environment. They also use knowledge of enemy/observer whereabouts. However, they only use current information regarding enemy positions and looking directions. This can be dangerous as enemies can quickly turn around, resulting in a completely different visibility map. Furthermore, the method requires global information about enemies (i.e. the current positions of all enemies are known to the virtual character, whether they are visible or not) as it does not consider previous knowledge about enemy whereabouts. In contrast, the covert pathfinding method proposed in this thesis uses past information regarding enemy positions, constructing a probability map of where enemies usually move about.

Marzouqi and Jarvis [MJ03] have done extensive work on covert pathfinding for robots. They use a regular discrete two-dimensional representation of the environment. In the case where no enemy information is available, a general visibility map is used. If enemy positions are known, the visibility map is created from the enemies' points of view. For the pathfinding, they use a distance transform algorithm to find the optimal path through the environment. In [MJ06] Marzouqi and Jarvis extend their work to allow for uncertain enemy positions. Uncertainty concerning the enemy position is introduced as a Gaussian distribution, representing the probability that the enemy has moved. The Gaussian distribution has been modified to take obstacles into account. The visibility map is calculated using the Gaussian distribution. Unlike the covert pathfinding method described in this thesis, their method does not include knowledge about previous enemy positions, but only considers the current positions of the enemies.

Park et al. [PCKL09] propose a road-map-based method for stealthy navigation. The aim of their method is for a robot to be able to intercept an enemy intruder, while trying to stay undetected at the same time. The method assumes that the predicted trajectory of the enemy intruder is known. The obstacles in the environment are represented as polygons and likewise the visibility calculations are done using polygonal isovists. The covert pathfinding method proposed in this thesis uses previous knowledge about enemy positions while this method uses the current enemy position as well as the predicted positions. Furthermore, this method only focuses on one enemy at a time.

Tews et al. [TSM04] suggest a method for multi-robot stealthy navigation based on waypoints and potential fields. The robots share information among themselves to improve the stealth. The problem with this method is the limitations of the potential field pathfinding. There is often a chance that the robot will end up in a local minimum and not be able to proceed. The proposed method only considers a single observer, while the covert pathfinding method suggested in this thesis can handle multiple enemies.

Mocholé et al. [MEJ+06] propose an emotional pathfinding algorithm that uses a mixture of particle simulation and graphs in order to do the pathfinding. The obstacles repel the particles and the goal attracts the particles. The pathfinding presents a set of possible paths, letting the final path be chosen by an emotional decision layer. Which path is chosen depends on the whereabouts of objects or entities that the character has emotional attachment to. Unlike the emotional pathfinding method proposed in this thesis, the work by Mocholé et al. does not use emotions for the actual pathfinding, but rather for choosing the proper path from several possible paths.

Donaldson et al. suggest a method [DPL04] for emotional pathfinding. However, most of their work focuses on hunger and thirst which are not emotions, but rather physiological states. When the "emotions" in their system are high, the heuristics of the A*-algorithm is more noisy, resulting in more unintelligent movements. Mostly, the emotions affect where the agent wants to go (which desire it wants to fulfill depending on whether it is hungry, thirsty or afraid of enemies), not so much how it gets there. Compared to the emotional pathfinding presented in this thesis, the algorithm proposed by Donaldson et al. uses emotions very little. Furthermore, no emotions attachment to geographical places are taken into account.

Olsen et al. [OHS08] present a system where emotions play a large role in real-time strategy games. While they do not use the emotions for pathfinding, but rather for improving strategy, they create emotion maps. These emotions maps are used to project emotional information onto other agents, hence implicitly sharing information about the environment. The emotions affect the decisions of the agents in simple ways depending on the type of character (for instance warrior or builder). In contrast, the emotional pathfinding proposed in this thesis utilizes the emotion maps, along with the current emotional state of the character, in the pathfinding.

5 Conclusions

This thesis focuses on creating believable virtual characters for games and other types of virtual worlds. Specifically, the work is divided into three parts; determining the necessary properties of a system to create believable characters using gameplay design patterns, introducing emotions into decision-making models, and finally improving pathfinding using visibility information and emotions.

This chapter briefly summarizes the contributions of this thesis. This is followed by a discussion and a section dedicated to ideas for future work.

5.1 Summary of Contributions

In this thesis work, several contributions have been made to the areas of believability, decision making and pathfinding. The full contributions included in this thesis are listed in Chapter 4 and in short they are the following:

- An agent architecture is described and analyzed in terms of believability. Gameplay design patterns are used to analyze the possibility of creating believable characters using the proposed architecture.
- EBNs are extended to include emotional impact. Emotions affect the activation spreading by affecting the parameters of the network and the probabilities of the effects. Furthermore emotions can affect the execution-value of each behavior module individually, targeting specific behaviors under certain emotional states. The goals of the network are completely emotional. The model by Loewenstein and Lerner [LWHW01] on how emotions affect decision making is used as an inspiration for this work.
- EmoBNs are extended to include the notion of time. By introducing time in EmoBNs, the character can reason about which action is best by taking into account how long it will take until the desired result is achieved. Moreover, emotional time-discounting is introduced to let the emotions of the character affect how it weighs delayed gratification against instant gratification.

- A comparison between EmoBNs and behavior trees with respect to functionality and management is conducted. The result suggests that behavior trees have less functionality but more desirable design and management qualities than EmoBNs.

- Emotions are introduced into behavior trees. The factors risk, time and planning are used together with the character's emotional state to affect the priority in a new type of priority selector; the emotional selector.

- Covert pathfinding is enhanced by using knowledge about previous enemy whereabouts. A probability map of enemy positions is created and used to calculate the visibility map using discrete isovist calculations. The calculation is more accurate from the character's point of view, since it can easily avoid the areas which it believes are visible to enemies. Furthermore, the visibility calculation is sped up considerably, because only a part of the probability map can be used to create the visibility map and still gain a good result. Moreover, a falloff function is used which further decreases the computation in large environments.

- Pathfinding is extended using the notion of emotion maps. One map is created per emotion. Each map represents the places where the character has felt this emotion, and to what extent. The character is guided through the environment based on its previous emotional state at the different positions. Its current emotional state affects how the maps are combined to affect the pathfinding.

5.2 Discussion

The work in this thesis involves extending decision-making methods to include emotional impact based on scientific studies conducted in the field of psychology. Using emotions in action selection mechanisms can give a richer character behavior if it is done properly. Many attempts have been made to make the facial and body animations of virtual characters more emotional. Also, plenty of work has been conducted to create appraisal mechanisms to trigger emotions. However, despite the apparent interest in emotional characters, little work has been done to incorporate emotions into the decision making. The contributions of this thesis with respect to emotional decision making are not meant to be viewed as a complete substitute for previous action selection methods. Instead, they are to be seen as a suggestion for how emotions could affect decision making in virtual characters. While the work on EmoBNs has been in progress for several years and has been used in real applications, the work on EmoBTs has yet to be fully explored, used, tested and extended. A long-term goal of the work in this thesis is to inspire the use of emotions in many different ways and in many different action selection mechanisms. Many game designers as well as researchers have their own preferred action selection mechanisms. The work on EmoBNs and EmoBTs is meant to inspire others to also incorporate emotions into their action selection mechanisms.

The results of the work presented in this thesis with respect to pathfinding show great promise. There exist many different pathfinding algorithms, but most of them use some sort of cost to measure how easily a character can move from one node to another. Using maps that affect this cost, and that hold different types of informative data, could be a means to get a more believable pathfinding behavior. Ideally, a variety of different influence maps will be used to create paths that appear natural to the player. What influence maps are used should vary greatly depending on the context. For instance, a character that is in a hurry to get somewhere should focus on finding the fastest path to the goal. In another situation, the character might be casually walking to the local store and could therefore use influence maps related to emotions and custom (i.e. in what areas does this character usually walk). The long-term goal behind the pathfinding work of this thesis is to move away from pathfinding that only focuses on finding the shortest path, and to move towards more natural-looking paths.

5.3 Future Work

The work in this thesis that focuses on creating emotional decision-making systems for virtual characters could greatly benefit from one or more user studies. Traditionally, evaluations of decision-making algorithms are made using statistical methods. For example, a football simulation is created and the teams are assigned different action selection mechanisms. The score of the game is used as an indication of how good the action selection mechanism is. When believability is the goal, however, one cannot rely on scores for measurement. After all, a very robotic and unrealistic character may perform very well in such a game. Instead, a user study would have to be conducted to analyze the believability of the models. To carry out such a study is not trivial, however. Many things will influence the user, graphics not least of all. Since the decision-making methods proposed in this thesis rely greatly on emotional influence, it is vital that the virtual characters are able to correctly display their emotions using body language and possibly facial expressions. During the course of this work, the project has switched rendering engines. Initially it was possible to blend animations together according to emotional state. This architecture had many problems, however, and needed to be replaced. The current system has no way of blending animations of the animated characters together. Beyond this, it is a matter of actually creating realistic animations, something which is non-trivial. A user study was indeed conducted using the Animalistic project. However, the results were inconclusive, showing mostly problems with other aspects of the installation. The study did show that it is difficult to perform a study on immersion, believability and interaction, because these areas can be rather fuzzy and are subject to large impact from incidental events. For instance, children failed to interact with the installation because they could not interpret what the animals were doing. Because of all the reasons mentioned above, the study of believability of the presented decision-making algorithms is left as future work.

As mentioned in Section 3.11, the biggest problems with EmoBNs are the tuning of parameters and the difficulty in debugging the network. While the bugs in the configurations are often easy to address, finding them can take a long time if the network is large. A visual debugging system would greatly improve debugging. It would also enable the game designer to get a better overview of the connections between the behavior modules, and the flow of activation. A real-time visualization tool for behavior networks would be a great asset to future research on behavior networks.

This thesis focuses mainly on fairly basic emotions such as fear, anger, and happiness. While the system fully supports more complex emotions such as guilt, gloating, and pride, these types of emotions require the ability to view events in a social context. The social component of the agent architecture is not implemented, but would be a valuable addition to the system. While there is a trust module that can represent the views of a character on another character, there is currently no way to evaluate the events in such a way that it would affect these trust values. Furthermore, there are many things linked to social behavior that would be of interest to this work, such as social group forming, shared goals, and social protocols.

The covert pathfinding proposed in Paper D uses only the probability distribution of enemy positions. It cannot incorporate more information into the map. For instance, if the character knows the exact whereabouts of one enemy, this should play a bigger part in its visibility calculation than the probability map. Furthermore, the character cannot reason about or detect cyclic patterns in the movements of the enemies. Let us assume that an enemy guard is patrolling an area in a circle-like path. A person would notice this pattern after a while and take advantage of the periods of time when it knows the guard will be unable to spot him/her. Enabling a virtual character to do the same could be beneficial but is difficult.

The downside of introducing more complexity is the inevitable increase in the number of parameters that need to be tweaked to get the desired behavior. Tuning the parameters and configurations of an EmoBN to get the desired behavior can be rather cumbersome at times, especially if the networks are fairly large and there is much sharing of effects and preconditions between behaviors. Ideally, one would like to automate the tuning of the parameters in some way. One alternative is applying learning to the behavior network in some way. During the course of this work, a master thesis project within the group was dedicated to trying to apply learning to behavior networks. While the results were interesting, they were nevertheless not directly usable. The main problem is the continuous environment the behavior network works within. The effects of an action are not immediate and it is therefore very difficult to determine the effect of an action. Many of the links suggested by the learning algorithm (a type of reinforcement learning) were due to incidental effects. Starting with an empty network, the learning algorithm managed to put together a behavior network that produced somewhat reasonable results. This in itself is excellent. However, the behavior was always better with a hand-tuned behavior network. As future work, it would be interesting to limit the learning in such a way that the structure of the network graph itself cannot be changed, only the parameters and effect probabilities. Furthermore it would be useful to introduce supervised learning in such a way that the behavior designer can specify general desired changes, e.g. “more of behavior A, less of behavior B in situation C”. This could potentially make is easier to tune large networks.

An interesting feature to introduce into the EmoBN is automatic affective appraisal. The behavior network should be able to detect the changes in the states, how desirable a state is and hence be able to trigger emotions automatically. Activation could possible be saved in each state and used as an indicator of the benevolence of each effect. Using techniques introduced in the master thesis on learning might prove successful in this, more controlled, evaluation problem.

This work has only just scratched the surface of emotional behavior trees. The proposed model could possibly be extended in many way. First of all it would be interesting to evaluate the performance of EmoBTs, see the drawbacks and the benefits, and expand it further. It would also be interesting to explore the possibility of creating a hybrid behavior network and behavior tree solution. For instance, it could be possible to have a behavior network that controls the abstract, high-level behaviors, while behavior trees are used to control the low-level behaviors. How such a model would function is yet unclear and is therefore a candidate for future work.

Bibliography

[ADP06] Ruth Aylett, Joao Dias, and Ana Paiva. An affectively driven planner for synthetic characters. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI press, 2006.

[Bar06] Lisa Feldman Barrett. Are emotions natural kinds? *Perspectives on Psychological Science*, 1(1):28–58, March 2006.

[Bat01] Michael Batty. Exploring isovist fields: space and shape in architectural and urban morphology. *Environment and Planning B: Planning and Design*, 28:123–150, 2001.

[BAZB02] N. Badler, J. Allbeck, Liwei Zhao, and M. Byun. Representing and parameterizing agent behaviors. In *Proceedings of Computer Animation*, pages 133–143, 2002.

[BBSS90] Herbert Bless, Gerd Bohner, Norbert Schwarz, and Fritz Strack. Mood and persuasion: A cognitive response analysis. *Personality and Social Psychology Bulletin*, 16(2):331–345, June 1990.

[BDDA94] Antoine Bechara, Antonio R. Damasio, Hanna Damasio, and Steven W. Anderson. Insensitivity to future consequences following damage to human prefrontal cortex. *Cognition*, 50(1-3):7–15, 1994.

[Beh04] Sven Behnke. Local multiresolution path planning. In Daniel Polani, Brett Browning, Andrea Bonarini, and Kazuo Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII*, volume 3020 of *Lecture Notes in Computer Science*, pages 332–343. Springer Berlin / Heidelberg, 2004.

[Ben79] M. L. Benedikt. To take hold of space: isovist fields. *Environment and Planning B: Planning and Design*, 6(1):47–65, 1979.

[BH05] Staffan Björk and J. Holopainen. *Patterns in Game Design*. Charles River Media, Hingham, 2005.

[Bow81] Gorden H. Bower. Mood and memory. *Cellular and molecular neurobiology*, 36(2):129–148, February 1981.

[CDHC10] Alex J. Champandard, Michael Dawe, and David Hernandez-Cerpa. Behavior trees: Three ways of cultivating game ai. Game Developers Conference, AI Summit, 2010.

[Cha07a] Alex J. Champandard. 10 reasons the age of finite state machines is over. www.aigamedev.com, 2007.

[Cha07b] Alex J. Champandard. Popular approaches to behavior tree design. http://aigamedev.com/open/article/popular-behavior-tree-design/, 2007.

[Cha07c] Alex J. Champandard. Understanding behavior trees. www.aigamedev.com, 2007.

[Cha08a] Alex Champandard. *Getting Started with Decision Making and Control Systems*, section 3.4, pages 257–264. Springer, 2008.

[Cha08b] Alex J. Champandard. Behavior trees for next-gen game ai. https://aigamedev.com/insider/presentations/behavior-trees/, 2008.

[Cha09] Alex J. Champandard. Behavior tree design patterns: Prioritization. www.aigamedev.com, 2009.

[CM92] Paul T. Jr. Costa and Robert R. McCrae. Normal personality assessment in clinical practice: The neo personality inventory. *Psychological Assessment*, 4(1):5–13, 1992.

[Dam95] Antonio R. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain*. Harper Perennial, 1995.

[Dam99] Antonio R. Damasio. *The Feeling of What Happens: Body, Emotion and the Making of Consciousness*. New York: Harcourt Brace, 1999.

[Dij59] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[Dor99a] Klaus Dorer. Behavior networks for continuous domains using situation-dependent motivations. In *Proceedings of IJCAI*, pages 1233–1238, 1999.

[Dor99b] Klaus Dorer. Extended behavior networks for the magma-freiburg team. RoboCup-99. Linköping University Press, 1999.

[Dor04] Klaus Dorer. Extended behavior networks for behavior selection in dynamic and continuous domains. In *Proceedings of the ECAI workshop Agents in dynamic domains*, 2004.

[DP05] João Dias and Ana Paiva. Feeling and reasoning: A computational model for emotional characters. In Carlos Bento, Amílcar Cardoso, and Gaël Dias, editors, *Progress in Artificial Intelligence*, volume 3808 of *Lecture Notes in Computer Science*, pages 127–140. Springer Berlin / Heidelberg, 2005.

[DPL04] Toby Donaldson, Andrew Park, and I-Ling Lin. Emotional pathfinding. In Ahmed Tawfik and Scott Goodwin, editors, *Advances in Artificial Intelligence*, volume 3060 of *Lecture Notes in Computer Science*, pages 31–43. Springer Berlin / Heidelberg, 2004.

[Ebb15] Hermann Ebbinghaus. *Memory: A contribution to experimental psychology.* Teachers college, Columbia university, 1885, translated reprint 1915.

[Ekm92] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992.

[Ekm99] Paul Ekman. *Facial Expressions*, chapter 16, pages 301–320. John Wiley and Sons, New York, 1999.

[EKMT03] Arjan Egges, Sumedha Kshirsagar, and Nadia Magnenat-Thalmann. A model for personality and emotion simulation. In Vasile Palade, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2773 of *Lecture Notes in Computer Science*, pages 453–461. Springer Berlin / Heidelberg, 2003.

[FAS04] Sarah Fischer, Kristen G. Anderson, and Gregory T. Smith. Coping with distress by eating or drinking: Role of trait urgency and expectancies. *Psychology of Addictive Behaviors*, 18(3):269–274, 2004.

[FG97] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. *Intelligent Agents III Agent Theories, Architectures, and Languages*, pages 21–35, 1997.

[Fri88] Nico H. Frijda. The laws of emotion. *American Psychologist*, 43(5):349–358, 1988.

[FSRE07] Johnny R.J. Fontaine, Klaus R. Scherer, Etienne B. Roesch, and Phoebe C. Ellsworth. The world of emotions is not two-dimensional. *Psychological Science*, 18(12):1050–1057, December 2007.

[Gan89] Richard M. Ganley. Emotion and eating in obesity: A review of the literature. *International Journal of Eating Disorders*, 8(3):343–361, 1989.

[GM01] Jonathan Gratch and Stacy Marsella. Tears and fears: modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01. ACM, 2001.

[GO07] Roland Geraerts and Mark Overmars. The corridor map method: A general framework for real-time high-quality path planning. *Computer Animation and Virtual Worlds*, 18(2):107–119, 2007.

[GS10] Roland Geraerts and Erik Schager. Stealth-based path planning using corridor maps. In *Computer Animation and Social Agents, CASA'2010*, 2010.

[GW97] Philip Goetz and Deborah Walters. The dynamics of recurrent behavior networks. *Adaptive Behavior*, 6(2):247–283, September 1997.

[Hec07] Chris Hecker. My liner notes for spore/spore behavior tree docs. www.chrishecker.com, 2007.

[HH06] Christopher K. Hsee and Reid Hastie. Decision and experience: why don't we choose what makes us happy? *Trends in Cognitive Sciences*, 10(1):31 – 37, January 2006.

[HJC03] Amy E. Henninger, Randolph M. Jones, and Eric Chown. Behaviors that emerge from emotion and cognition: implementation and evaluation of a symbolic-connectionist architecture. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 321–328. ACM, 2003.

[IDN87] Alice M. Isen, Kimberly A. Daubman, and Gary P. Nowicki. Positive affect facilitates creative problem solving. *Journal of Personality and Social Psychology*, 52(6):1122–1131, 1987.

[IJMR85] Alice M. Isen, Mitzi M. Johnson, Elizabeth Mertz, and Gregory F. Robinson. The influence of positive affect on the unusualness of word associations. *Journal of Personality and Social Psychology*, 48(6):1413–1426, June 1985.

[Isl05] Damian Isla. Handling complexity in the halo 2 ai. Game Developers Conference, 2005.

[Isl08] Damian Isla. Building a better battle - the halo 3 ai objectives system. Game Developers Conference (talk), 2008.

[KBT03] M. Kallmann, H. Bieri, and D. Thalmann. Fully dynamic constrained delaunay triangulations. *Geometric Modelling for Scientific Visualization*, 4:74–123, 2003.

[Kha86] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, March 1986.

[KM95] Kris N. Kirby and Nino N. Maraković. Modeling myopic decisions: Evidence for hyperbolic delay-discounting within subjects and amounts. *Organizational Behavior and Human Decision Processes*, 64(1):22–30, October 1995.

[KMP00] Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular norms*, volume 8. Springer, 2000.

[Kna11] Bjoern Knafla. Introduction to behavior trees. http://bjoernknafla.com/introduction-to-behavior-trees, March 2011.

[KS03] Kris N. Kirby and Mariana Santiesteban. Concave utility, transaction costs, and risk in measuring discounting of delayed rewards. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(1):66–79, January 2003.

[LB07a] Petri Lankoski and Staffan Björk. Gameplay design patterns for believable non-player characters. In Baba Akira, editor, *Situated Play: Proceedings of the 2007 Digital Games Research Association Conference*, pages 416–423, Tokyo, September 2007. The University of Tokyo.

[LB07b] Petri Lankoski and Staffan Björk. Gameplay design patterns for social networks and conflicts. In *Computer Game Design and Technology Workshop, John Moores University, Liverpool*, pages 76–85, 2007.

[LB08] Petri Lankoski and Staffan Björk. Character-driven game design: Characters, conflicts, and gameplay. In *Sixth International Conference in Game Design and Technology*, 2008.

[LBP97] Mary Frances Luce, James R. Bettman, and John W. Payne. Choice processing in emotionally difficult decisions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(2):384–405, 1997.

[LD04] Fabrice Lamarche and Stéphane Donikian. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum*, 23(3):509–518, 2004.

[LeD96] Joseph E. LeDoux. *The Emotional Brain.* New York: Simon and Schuster, 1996.

[LeD00] Joseph E. LeDoux. Emotion circuits in the brain. *Annual Review of Neuroscience*, 23(1):155–184, 2000.

[LeD03] Joseph E. LeDoux. The emotional brain, fear, and the amygdala. *Cellular and molecular neurobiology*, 23(4):727–738, 2003.

[LK00] Jennifer S. Lerner and Dacher Keltner. Beyond valence: Toward a model of emotion-specific influences on judgement and choice. *Cognition and Emotion*, 14:473–493, 2000.

[LK01] Jennifer S. Lerner and Dacher Keltner. Fear, anger, and risk. *Journal of Personality and Social Psychology*, 81:146–159, 2001.

[LL03] George Loewenstein and Jennifer S. Lerner. *The Role of Affect in Decision Making*. Oxford University Press: New York, 2003.

[LM07] Christine L. Lisetti and Andreas Marpaung. Affective cognitive modeling for autonomous agents based on scherer's emotion theory. *KI 2006: Advances in Artificial Intelligence*, pages 19–32, 2007.

[Loe96] George Loewenstein. Out of control: Visceral influences on behavior. *Organizational Behavior and Human Decision Processes*, 65(3):272–292, 1996.

[Loe10] George Loewenstein. Insufficient emotion: Soul-searching by a former indicter of strong emotions. *Emotion Review*, 2(3):234–239, July 2010.

[LPW79] Tomás Lozano-Pérez and Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, October 1979.

[LWHW01] George Loewenstein, Elke U. Weber, Christopher K. Hsee, and Ned Welch. Risk as feelings. *Psychological bulletin*, 127(2):267, 2001.

[Mae89] Pattie Maes. How to do the right thing. *Connection Science Journal*, 1(3):291–323, 1989.

[MEJ+06] José Mocholí, José Esteve, Javier Jaén, Raquel Acosta, and Pierre Xech. An emotional path finding mechanism for augmented reality applications. In Richard Harper, Matthias Rauterberg, and Marco Combetto, editors, *Entertainment Computing - ICEC 2006*, volume 4161 of *Lecture Notes in Computer Science*, pages 13–24. Springer Berlin / Heidelberg, 2006.

[MF09] Ian Millington and John Funge. *Artificial Intelligence for Games*, chapter 6. Morgan Kaufmann, 2 edition, 2009.

[MG02] Stacy Marsella and Jonathan Gratch. A step toward irrationality: using emotion to change belief. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, AAMAS '02, pages 334–341. ACM, 2002.

[Mil06] Ian Millington. *Artificial Intelligence for Games*. Morgan Kaufmann, 1 edition, 2006.

[MJ03] M. Marzouqi and R.A. Jarvis. Covert robotics: Covert path planning in unknown environments. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2003.

[MJ06] Mohamed S. Marzouqi and Ray A. Jarvis. New visibility-based path-planning approach for covert robotic navigation. *Robotica*, 24:759–773, August 2006.

[NE09] Randolph M. Nesse and Phoebe C. Ellsworth. Evolution, emotions, and emotional disorders. *American Psychologist*, 64(2):129–139, 2009.

[OHS08] Megan Olsen, Kyle Harrington, and Hava Siegelmann. Emotions for strategic real-time systems. *AAAI Emotion, Personality, and Social Behavior Technical Report*, pages 104–110, 2008.

[OKG08] Mark Overmars, Ioannis Karamouzas, and Roland Geraerts. Flexible path planning using corridor maps. *Algorithms - ESA 2008*, pages 1–12, 2008.

[PA05a] Hugo C. Pinto and Luis O. Alvares. An extended behavior network for a game agent: An investigation of action selection quality and agent performance in unreal tournament. In *Proceedings of MICAI 2005: Advances in Artificial Intelligence*, volume 3789, pages 287–296, 2005.

[PA05b] Hugo C. Pinto and Luis O. Alvares. Extended behavior networks and agent personality: Investigating the design of character stereotypes in the game unreal tournament. In Themis Panayiotopoulos, Jonathan Gratch, Ruth Aylett, Daniel Ballin, Patrick Olivier, and Thomas Rist, editors, *Intelligent Virtual Agents*, volume 3661 of *Lecture Notes in Computer Science*, pages 418–429. Springer Berlin / Heidelberg, 2005.

[PB08] Hans-Rüdiger. Pfister and Gisela Böhm. The multiplicity of emotions: A framework of emotional functions in decision making. *Judgment and Decision Making*, 3(1):5–17, January 2008.

[PCKL09] Jung-Hee Park, Jeong-Sik Choi, Jimin Kim, and Beom-Hee Lee. Roadmap-based stealth navigation for intercepting an invader. In *IEEE International Conference on Robotics and Automation (ICRA'09)*, pages 442–447. IEEE, 2009.

[Pic97] Rosalind W. Picard. *Affective Computing*. MIT Press, 1 edition, 1997.

[PT96] Anne L. Powell and Mark H. Thelen. Emotions and cognitions associated with bingeing and weight control behavior in bulimia. *Journal of Psychosomatic Research*, 40(3):317–328, 1996.

[RP99] Rajagopal Raghunathan and Michel Tuan Pham. All negative moods are not equal: Motivational influences of anxiety and sadness on decision making. *Organizational Behavior and Human Decision Processes*, 79(1):56–77, 1999.

[SAGA00] Eric Stice, Donna Akutagawa, Amit Gaggar, and Stewart Agras. Negative affect moderates the relation between dieting and binge eating. *International Journal of Eating Disorders*, 27(2):218–229, 2000.

[SC83] Norbert Schwarz and Gerald L. Clore. Mood, misattribution, and judgments of well-being: Informative and directive functions of affective states. *Journal of Personality and Social Psychology*, 45(3):513 – 523, 1983.

[SC03] Norbert Schwarz and Gerald L. Clore. Mood as information: 20 years later. *Psychological Inquiry*, 14(3-4):296–303, 2003.

[SC07] Norbert Schwarz and Gerald L. Clore. Feelings and phenomenal experiences. *Social psychology: Handbook of basic principles*, pages 385–407, 2007.

[Sch00] Norbert Schwarz. Emotion, cognition, and decision making. *Cognition and Emotion*, 14(4):433–440, 2000.

[Sch09] Klaus Scherer. The dynamic architecture of emotion: Evidence for the component process model. *Cognition and Emotion*, 23(7):1307–1351, November 2009.

[SMM00] Paul Slovic, John Monahan, and Donald G. MacGregor. Violence risk assessment and risk communication: The effects of using actual cases, providing instruction, and employing probability versus frequency formats. *Law and Human Behavior*, 24(3):271–296, 2000.

[SPFM05] Paul Slovic, Ellen Peters, Melissa L. Finucane, and Donald G. MacGregor. Affect, risk, and decision making. *Health Psychology*, 24(4):S35–S4, July 2005.

[SS02] Robert C. Solomon and Lori D. Stone. On "positive" and "negative" emotions. *Journal for the Theory of Social Behaviour*, 32(4):417–435, 2002.

[Ste95] Anthony Stentz. The focussed D* algorithm for real-time replanning. In *In Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[TBB01] Dianne M. Tice, Ellen Bratslavsky, and Roy F. Baumeister. Emotional distress regulation takes precedence over impulse control: If you feel bad, do it! *Journal of Personality and Social Psychology*, 80(1):53–67, 2001.

[Toz01] Paul Tozour. *Game programming gems 2*, chapter 3.6. Cengage Learning, 2001.

[TSM04] A.D. Tews, G.S. Sukhatme, and M.J. Mataric. A multi-robot approach to stealthy navigation in the presence of an observer. In *Proceedings of IEEE International Conference on Robotics and Automation, (ICRA'04)*, volume 3, pages 2379–2385. IEEE, 2004.

[TWB07] Emmanuel Tanguy, Philip Willis, and Joanna J. Bryson. Emotions as durative dynamic state for action selection. In *In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad*, pages 1537–1542. Morgan Kaufmann, 2007.

[Tyr94] Toby Tyrrell. An evaluation of Maes's bottom-up mechanism for behavior selection. *Adaptive Behavior*, 2:307 – 348, March 1994.

[vBS08] Arthur van Bilsen and Egbert Stolk. Solving error problems in visibility analysis for urban environments by shifting from a discrete to a continuous approach. In *International Conference on Computational Sciences and Its Applications, ICCSA'08.*, pages 523–528, 2008.

[Vel97] Juan D. Velásquez. Modeling emotions and other motivations in synthetic agents. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, AAAI'97/IAAI'97, pages 10–15. AAAI Press, 1997.

[WG03] Timothy D Wilson and Daniel T Gilbert. Affective forecasting. volume 35 of *Advances in Experimental Social Psychology*, pages 345 – 411. Academic Press, 2003.

[WP08] Marc Wittmann and Martin P. Paulus. Decision making, impulsivity and time perception. *Trends in Cognitive Sciences*, 12(1):7 – 12, 2008.

[YC05] Chan-Woo Yu and Jin-Young Choi. Behavior decision model based on emotion and dynamic personality. In *Proceedings of International Conference on Control, Automation and Systems (ICCAS)*, pages 101–106, 2005.

[YRFB99] Michelle S. M. Yik, James A. Russell, and Lisa Feldman Barrett. Structure of self-reported current affect: Integration and beyond. *Journal of Personality and Social Psychology*, 77(3):600–619, 1999.