Utrecht University
Information and Computing Sciences
Game and Media Technology Department

Emotion in Computer Games

by

Mate Tomin

1st supervisor: dr. ir. Arjan Egges
2nd supervisor: dr. ir. Robbert-Jan Beun

Academic year 2012-2013

Contents

Chapter 1

Introduction

Over the past years, the game industry has made numerous impressive advances due to the constantly increasing capabilities of modern computer hardware, particularly the processors and graphical accelerators. For example, the physical realism of recent video game characters has shown some incredible improvements (Figure 1.1). New hardware technologies have also radically changed the player's experience of video games. For instance, smart phones enable users to play their favourite video games anywhere and anytime (Figure 1.2a). Furthermore, the release of motion detection devices has created an entirely new experience of interaction. Players need no controllers to play games anymore (Figure 1.2b).

Figure 1.1: A football player (Lionel Messi) celebrates his goal in Pro Evolution Soccer 2012

These examples show that recent video games are no longer just a few minutes amusement. Realistic graphics, new playing experience and easy control have made computer games similar to interactive films, capable of engaging players for hours. However, players have great eyes for detecting even the tiniest flaws in games, especially in game characters, which may immediately disrupt player's engagement. Thus, giving life to believable non-playing characters (NPCs) in games is one of the most difficult challenges that video game industry faces. Fortunately, recent technology can provide detailed and realistic appearance, motion and intelligent behaviour. However, this is not always enough, in


(a) (b)

Figure 1.2: Sony Xperia smartphone specifically made for playing games and Kinect device that detects players' motion.

order to make players better relate, understand and believe in NPCs, affective realism should be also present in games. Research shows that the expression of emotions is important to create an illusion of life in video game characters and to increase their believability [34].

The simplest way to provide affective realism is to endow characters with the capability of visually displaying emotion expressions via gestures, body language, facial expressions, etc. Normally, the visual representation of emotion expressions follows a scripted behaviour: emotions are preprogrammed. For example, if a character loses then it will be programmed to display sadness. Let us call such approach, where events are directly linked to emotional expressions, a *black-box model.* In the case of simple games, the number of actions that NPCs can take is relatively low. For that reason, the *black-box* model is a great choice because designers can just assign events to emotion expressions.

In a more complex game environment, in which player's actions can directly affect the progression of narratives, game designers will not be able to predict precisely the direction taken by the story anymore, as they do not know what players will do. In such a game, using a *black-box* model to elicit emotions in NPCs becomes infeasible because it would require an extensive amount of work to assign emotions to all the possible events that may occur throughout the game. A better solution for such an interactive environment is to develop characters that are capable of evaluating information from their environment and generating a fitting emotional response. In other words, to enable NPCs to identify the emotional meaning of the situations by themselves. We call this *appraisal-based* model. In our opinion, implementing an *appraisal-based* model has the advantage that it is more generic, which implies that it would require less ad-hoc solutions to generate emotions.

1.1 Objectives

Computer scientists normally divide emotions into three stages (Figure 1.3) [33]. The first phase introduces how an emotion is generated: the situation is appraised, and based on that, an emotion is elicited. The second phase describes how certain intensity values are assigned to the triggered emotion, so the emotion can be truly experienced. The third phase depicts how this experienced emotion may affect decision making and behaviour, which will subsequently

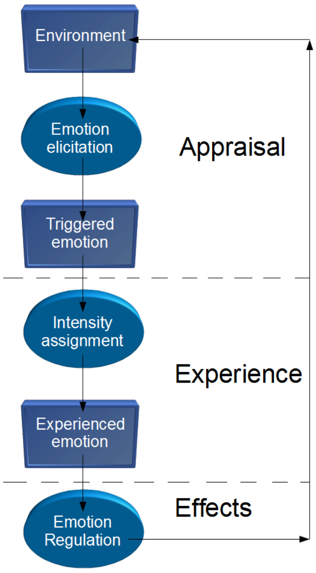


Figure 1.3: This figure introduces a general view of emotion in terms of data (square boxes) and processes (rounded boxes).

impact the game environment as well. It should be noted, however, that not all the researchers share the idea of applying all the three stages. For example, Hudlicka proposes a simpler view, which does not separate experiencing emotion from the other two [14].

Any of these stages would form a sufficient topic for an in-depth research. In our case, we focus exclusively on the first stage, with prime objective to generate emotions. Investigation on intensity assignment and the effect of emotions on NPC'c can be done in future research, making this work complete.

Our goal is to develop a generic model for triggering emotions in game characters. Our hypothesis is that such a model can produce emotions at least as well as *black-box* models, while having the advantage of not requiring ad-hoc solutions. Furthermore, we expect that the dynamic nature of our model will elicit a richer set of emotions. To give an example, if a character loses the game, then such event could be configured to trigger rage in the NPC by the *black-box* model approach. However, there are more emotions that could be appropriate for this particular situation, such as shame or anger, depending on the subjective interpretation of the current game state.

We also need to better understand emotions. Fortunately, they have been the subject of psychological research for many years. However, researchers have not been able to come to consensus regarding the origin and the reason for the existence of emotions. Consequently, many theories have been developed over the years, often representing contrasting views. The major theories will be introduced in Chapter 2, without any intention to judge why one theory would be better than another.

We break down our main goal into three sub-goals. Firstly, we need a computer game, in which we can evaluate our *appraisal-based* model. Our goal is not to invent a new game, but rather to find one that has a simple environment and limited number of actions that a player can take. A game with such features would be perfect for our evaluation, because we can keep track of all the changes in the game. In Chapter 3, we will present all the details about this game such as its rules, environment, game play and strategies. The reason for introducing this game at the beginning of the thesis is that, this way, we will be able to provide examples.

Secondly, we need to find an emotion theory, on which we can base our research. In our opinion, appraisal theories are the most suitable for our purpose because they approach emotions from a subjective point of view: they state that events do not have significance in themselves, what matters is how individuals interpret them. This concept perfectly fits in our research. Thus, we have decided to base our research on a popular appraisal theory developed by Klaus Scherer [30, 31]. His theory is detailed and highly regarded by psychologists. Furthermore, there have already been attempts to implement his work by computer scientists [19, 6, 7, 27, 18].

Adopting a psychological theory such as Scherer's for implementation purposes will raise a few issues. Appraisal theories approach emotions from a theoretical point of view: theories are too detailed and do not have any information on how to implement them. Such a theory requires simplification of parts that are too detailed and extending the theory when not enough information is provided. In Chapter 4, we give a short overview on the most essential parts of Scherer's theory. Furthermore, we introduce our computational model, which is used to trigger emotions.

Thirdly, we need our NPC to be able to evaluate changes in the game environment, so based on its evaluation, our computational model will automatically trigger an emotion. Thus, we implement our NPC as an intelligent agent, so its mental state is the explicit representation of the game world. The BDI (Belief-Desire-Intention) theory is commonly used to describe mental states of agents in terms of their beliefs, desires and intentions [5]. Thus, we think it would be an interesting idea to link Scherer's theory to a BDI agent, since according to appraisal theories, people appraise situations respect to their goals, beliefs, values and standards [14]. In Chapter 5, we illustrate all the details on this subject.

In Chapter 6, we present the implementation part of our research. Next, we evaluate our *appraisal-based* model and present our findings (Chapter 7). Finally, we draw our conclusions (Chapter 8) and discuss possible future work (Chapter 9).

Chapter 2

Related Work

As a start-off, we try to get a better understanding of the emotions by studying existing psychological research. In this chapter, we give an overview of several emotion theories and present various game-related research that have adapted Scherer's theory for their research.

2.1 Emotion theories

Aristotle spent a great deal of time attempting to understand human emotions. His work is particularly important because his book *Rhetoric* is considered to be the first known cognitive approach to examining emotion in Western history [23]. In his book, he provides an interesting analysis of emotions, in which he breaks down emotions into what beliefs they presuppose (e.g. anger may require the belief that someone is doing something wrong), their valence (e.g. anger is unpleasant), their associated actions (e.g. anger urges individuals to take action), and their cognitive aspect as well (e.g. anger influences decision making). It is interesting to see that Aristotle was already able to define different criteria to distinguish emotions.

The first modern book on emotion was written by Charles Darwin, called *The Expression of the Emotions in Man and Animals*, in the 19th century [9]. Darwin assumes that all emotions have specific features such as facial, physiological and behavioural responses. He believed that emotional expressions were shaped through evolution, and only those that remained had survival values.

At the same time, another interesting theory was developed by James and Lang independently [15, 16]. Their approach reverses the way humans think of emotion elicitation. James and Lang state that emotions are genetic reflexes that need no intervention of brain. The nervous system automatically creates physiological changes such as muscular tension and rise in heart rate, and these changes will trigger emotions, rather than be their cause. In other words, people do not cry because they are feeling sad, rather they feel sad because they are crying (Figure 2.1). This approach has been criticized broadly for its reverse thinking and, thus, is disregarded by many researchers.

In the early 1970s, Ekman further investigated Darwin's idea about emotions being developed during evolution: he started examining facial expressions to find out whether there are emotions that can be found in all humans, regardless of

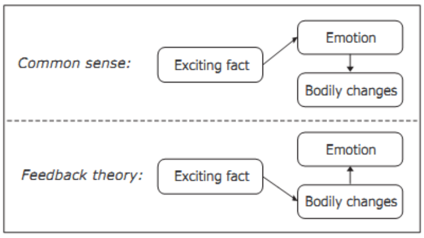


Figure 2.1: James's reversal of common sense and his "feedback theory" (from [33])

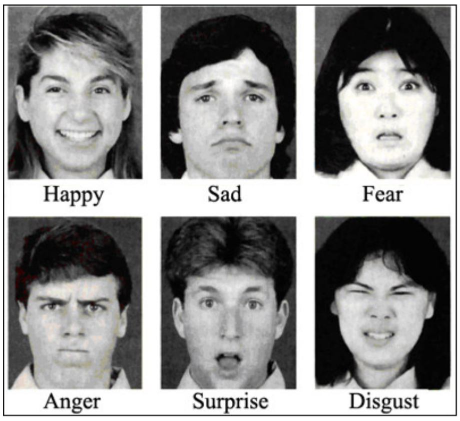


Figure 2.2: Ekman and Friesen (1971) concluded that there are six basic emotions shared by all humans: happiness, sadness, fear, anger, surprise, and disgust.

ethnic and cultural differences [12]. He found that there is a limited repertoire of basic, homogeneous emotions with highly prototypical characteristics such as fear, disgust, anger, sadness, surprise and happiness (Figure 2.2). Based on his findings, the so-called discrete emotion theory was born. Discrete emotion theories state that a limited number of core emotions exist (e.g. Ekman's theory) and the combination of these basic emotions will result in more. Opposed to this statement, other emotion theories such as dimensional or appraisal do not recognize the existence of basic emotions but rather posit a continuous multi-dimensional emotion space, in which singular emotions can be defined. These theories will be discussed in the next sections.

In the beginning of 1990s, the neuroscience field has also started investigating emotions. Its findings have radically changed people's thinking about emotions. The study has shown that there is a necessity of emotions in people's lives: people without emotions will not be "super-rational" as expected, rather they will be incapable of making sensible social decisions [8]. Sometimes, emotions can be disadvantageous to reasoning, but a life without emotions would be far worse for decision making. Furthermore, Damasio proposes that emotions should be divided into two types: primary and secondary. The reason is that primary emotions are innate, developed during phylogeny to support fast and reactive response behaviour. Secondary emotions are assumed to arise from higher cognitive processes, based on ability to evaluate preferences over outcomes and expectations.

It can be clearly seen that there are a lot of different approaches, and thus, disagreements on emotions. Computer scientists normally prefer one of the following theories.

2.1.1 Dimensional theories

In dimensional theories, emotions are represented as single points in a space where each dimension is seen as a specific property. These theories are attractive for practical implementation because they provide a way of describing emotional state that is more tractable than using words. This way, dimensional descriptions can be translated easily into and out of verbal descriptions. There are many variations of dimensional theories that mainly differ in the number and names of dimensions.

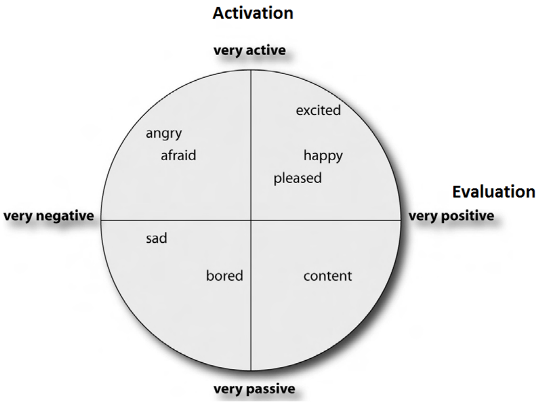


Figure 2.3: The 2D representation of emotions (X-axis indicates evaluation and Y activation (from [11]))

In two-dimensional representation of emotions, axes are named evaluation (from negative to positive) and activation (from passive to active)(Figure 2.3)[32]. These dimensions can be used to describe general emotion tendencies, including low-intensity emotions.

However, Mehrabian argues that two-dimensions are not always enough and develops a new theory, in which he postulates at least three dimensions to represent emotions without ambiguity [21]. His work uses pleasure, arousal and dominance dimensions (Figure 2.4), in which pleasure and arousal dimensions correspond to evaluation and activation, respectively, of the 2D representation. He considers dominance as the third dimension which is essential, because otherwise there would be no way of differentiating certain emotional states such as fear and anger. They are both unpleasant emotions, but anger is a dominant, whereas fear is a submissive emotion.

The advantage of dimensional approaches is that they can be easily linked to different emotion theories and also offer an easy way to represent emotions. Becker proposes that dimensional theories seem to be a great choice for trig-

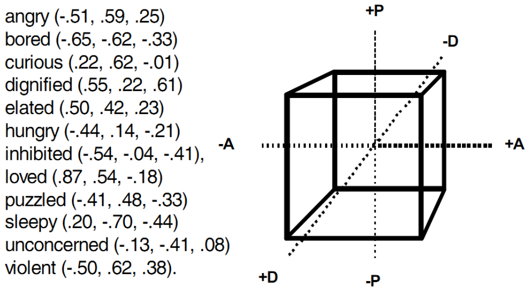


Figure 2.4: The Pleasure-Arousal-Dominance (PAD) dimensional theory

gering emotions in video game characters [4]. Nevertheless, the majority of psychologists do not find these theories sufficient because they think that using 2 or 3 dimensions results in loss of information due to simplifications and generalizations.

2.1.2 Appraisal theories

Appraisal theories are even more detailed and sophisticated than dimensional theories. Arnold was the first researcher who introduced the concept of appraisal, stating that events do not have significance in themselves, what matters is how individuals interpret them [2]. For example, the outcome of a football match might elicit happiness, sadness or indifference in a fan, depending on how this person appraises the event.

Following Arnold's seminal work, Lazarus distinguishes two processes that allow an individual to stabilize his relation with the environment: cognitive evaluation (appraisal) and adaptation (coping). The first evaluates the relevance and congruence of a stimulus in relation to the individual's well-being, whereas the second evaluates whether there are available resources to cope with this stimulus [17].

The most popular and wide-spread appraisal-based approach was proposed in the book of Ortony, Clore and Collins and named the OCC model after the names of the authors [26]. The theory provides a description of 22 emotions, in which emotions are valenced reactions to three types of stimuli in the hierarchy: events, agents and objects (Figure 2.5). These types make the model a good choice for practical work in Artificial Intelligence systems because they provide a clear and convincing structure of the eliciting conditions of emotions.

The consequences of events can be appraised as desirable, undesirable regarding one's goals. For instance, joy about winning a computer game is an event-based emotion. The satisfaction of having played through a game is a desirable consequence of the event of winning the computer game. If one focuses on an action of an agent, the action can be appraised as praiseworthy or blameworthy regarding one's standards. For example, in games, a player may experience pride, if they succeed in escaping from a death-threatening battle because of their smart decision making. If one focuses on an aspect of an object, one can appraise this aspect as liking or disliking with respect to one's attitudes. For example, love for an old car is an object-based emotion, because

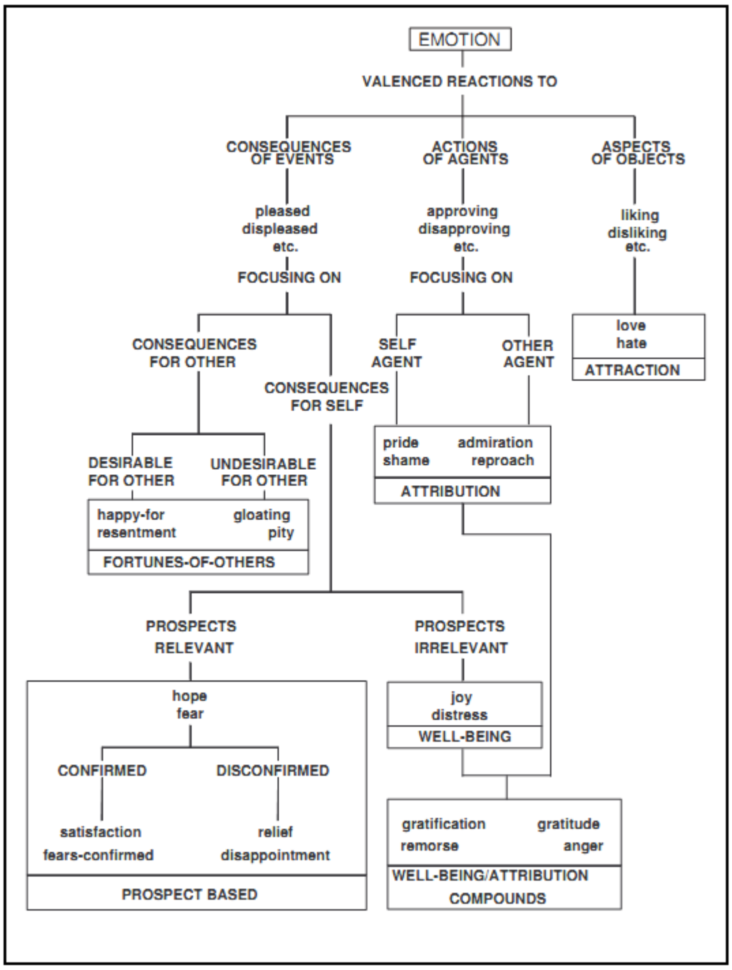


Figure 2.5: The OCC-model of emotions (from Ortony et al. 1988, p. 19)

the car may have appealing aspects according to one's attitudes.

It is important to note that the entire cognitive state of an agent is never described by emotions because emotions are always relative to individual objects, actions and events. Thus, an agent can be happy about event X but sad about Y simultaneously.

Scherer gives a completely different definition, approaching emotions from another perspective. According to him, emotion is a relatively brief episode on synchronized responses of most or all organic systems for the evaluation of an external or internal event [30, 31]. He has published a considerable amount of emotion-related literature, but our focus is mainly on his Component Process Model (CPM), which investigates our main interest, namely the appraisal process of emotions. The first version of this model was published approximately 30 years ago, and it has been refined many times without changing its underlying principles. In line with the literature, we will refer to the work of Scherer's Component Process Model simply as the CPM model henceforth.

According to Scherer, the type of emotions can be determined by the result of an evaluation or appraisal of an event in terms of its significance for the survival and well-being of the person. Scherer suggests a set of criteria that is considered to be the necessary minimum to adequately evaluate emotion-producing stimuli. Interestingly, as opposed to Arnold, Scherer defines 4 sequential phases (Stimulus Evaluation Checks (SECs)) for producing emotions:

- How relevant is this event for the individual? Does it directly affect herself or her social reference group? (Relevance detection)
- What are the implications or consequences of this event and how do these affect the individual's well-being and immediate or long-term goals? (Implication assessment)
- How well can the individual cope with or adjust to these consequences? (Coping potential)
- What is the significance of this event with respect to the individual's self-concept and to social norms and values? (Normative significance)

The 4 phases can be even further divided into different numbers of so-called sub-checks. It is important to stress that the outcomes of SECs are subjective, so they will be exclusively depending on the appraising individual. Sub-checks are also known as *appraisal variables*. Henceforth, we will use the term appraisal variables instead of sub-checks because that term is used more frequently in literature.

2.2 Scherer's theory in practical use

In this section, our main interest is to investigate existing literature, which has adopted Scherer's work. The literature can be separated into three research groups: creating believable expressions, formalizing psychological theories and developing computational models.

2.2.1 Believable emotional expressions

This group investigates how to create believable and natural emotional expressions for virtual characters. This question has been studied by many researchers but only a few of them adopts Scherer's findings. According to Pelachaud, in most virtual characters, emotions appear as full-blown expressions following a trapezoidal temporal course, which means that expressions appear, remain and disappear [28]. This is not natural for the critical eyes of the game players. Scherer's research suggests that expressions of emotions arise from the sequence of facial actions. He also provides details on this by specifying the names of participating facial muscles and the length of their movements [30, 31]. There have been attempts to adapt these guidelines in virtual characters in order to see whether they are indeed more believable [27, 18, 28] (Figure 2.6). They all concluded that sequential emotional expressions are found to be more believable. Unfortunately, this group does not have much to offer for our research, because its main focus is on displaying emotions.

Figure 2.6: Sequential representation of fear in a virtual character (from [27])

2.2.2 Formalizing emotions

The second group attempts to formalize psychological models of emotion in a complete and rigorous manner. These formalizations attempt to capture the logical structure underlying the emotion theories. Majority of researchers formalize the OCC model in logic because its structure is the most suitable for such practical works [22, 33, 25, 24]. There has been no research on formalizing Scherer's theory in logic yet and we have no intention to do that either. However, this group might provide some useful information, which can give us guidance to link our computational model to an intelligent agent.

2.2.3 Computational models

The last group has its focus on developing computational models, which normally specify the triggering conditions of emotions. The problem is that the complete implementation of the psychological theories would require a precise simulation of a human.

One of the most important research projects in this field has been undertaken by Gratch & Marsella [13]. They have developed a domain-independent framework, which creates a general computational model of the underlying mechanism of human emotion based on Lazarus's work. The goal of their model is to explain how emotions might arise from an agent's mental state and subsequently impact decision-making. However, their work provides only guidelines and lacks formal definitions. Interestingly, they suggest that the appraisal process should be linked to cognition, similarly as we are intending to do.

Marinier et al. take a different different approach and attempt to integrate emotions into cognition. More specifically, they integrate Scherer's CPM with a theory of cognitive control PEACTIDM [19]. PEACTIDM can be decomposed into sets of abstract functional operations hypothesized as the building blocks of immediate behaviour such as Perceive, Encode, Attend, Comprehend, Tasking, Intend, Decode, and Motor (Figure 2.7). Mapping appraisal variables to emotions and intensity calculation are both demonstrated and also evaluated in a PAC-MAN like environment. Furthermore, they add other affective components to their research such as feeling and mood to make sure that transition between emotional states occur smoothly.

Courgeon et al. have developed an application that generates appraisal events during a real-time interaction with a user [7]. In their work, a virtual character is used to display the triggered emotion such as joy, sadness, anger and guilt. They define Gaussian curves to enable appraisal variables evaluation on a continuous scale. According to them, this way the computation becomes less restricted. They present a scenario in their board game environment, which

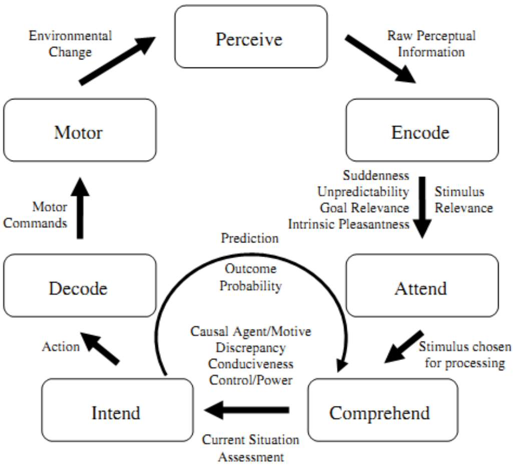


Figure 2.7: The basic PEACTIDM cycle with its corresponding appraisal variables from [19])

.

explains how emotions are precisely triggered.

Broekens et al. provide a more formal research than the previous researchers. They have made an attempt at narrowing the gap between appraisal theories and computational models by proposing a formal notation for the declarative semantics of the structure of appraisal [6]. They claim that such formalism helps to design and evaluate computational models as well as the integration of appraisal theories. In their work, to prove that formal specification helps with evaluation, they integrate two appraisal theories and use it as a computational model, which is subsequently embedded in an emotional agent using a PAC-MAN environment.

Chapter 3

Game

In this chapter, we propose a game that we can use later for testing our *appraisal-based* model. We investigate which game types need emotions, and based on our findings, we will develop a game and present its environment, gameplay, and strategies.

3.1 Emotion in various game genre

There are many types of games but not all of them need emotional content included. Adams & Rollings discuss the role of emotions in various game genres in their book [29]. In this chapter, our goal is to find and develop a simple game that could be used for evaluating our *appraisal-based* model. Thus, we investigate the findings of Adams & Rollings and, based on that, we propose a game.

According to them, strategy and action games have almost no emotional content. One would think that emotions are present in Role-playing games (RPGs) because of their deep stories. However, the authors think that emotions tend to get lost in the extensive bookkeeping in RPGs because players mainly focus on selling and buying items as well as developing character's skills, which often result in obscuring emotional content.

If games do not focus on strategy and high-speed actions then it is more likely that players can enjoy the emotionally enriched world. This means that affective characters are more likely to play a big role in adventure games.

Board and card games do not require much of emotional content but there are certain games that may benefit from using them for strategical purposes. For example, a poker player that attempts to hide or exaggerate his emotions in order to mislead their opponents.

Simulation games normally describe real life situations. This type of game definitely requires emotions. For example, it would be really odd to play *The Sims* without seeing characters expressing their emotions in real-life situations.

An interesting new genre (serious games) entered the game industry a couple of years ago. This game type simulates real-life situations in virtual environments. Obviously, people do not seem real without being endowed with emotions. This means that such environments must have realistic emotional content included.

Existing literature has also adopted at least some part of Scherer's theory in various games such as a card game [3, 4], a board game [7], a Pac-Man environment [6] or a modified version of Pac-Man called Easter [20].

When making our final choice about which game to develop, we also have to keep in mind an important matter. How do we know when to trigger an emotion during the game? Using discrete time instead of continuous would resolve this issue because, that way, emotions are only triggered at particular points in time. Discrete time can be used to closely approximate the action in board and card games, especially those that are played in a turn-based system. Thus, we have made the decisions to develop such a game type, more specifically the popular game, called Battleship.

3.2 Battleship

Battleship is a guessing game, which is known as a pencil and paper game throughout the world. The goal of the game is to destroy all your opponent's ships before your opponent destroys your own. The game is played by two players, on a no time-limit 2 player turn-based system. As it is a turn-based game, there is a clear separation between the game flow and thinking process. This is beneficial because the agent's deliberation process will never be interrupted. Henceforth, agents and NPC terms will be used interchangeably in this thesis.

3.2.1 Environment

The computer game version of Battleship is played on two grids, one for each player. The grids are normally square shaped consisting of 10 x 10 individual squares. However, it can vary from fewer to more squares depending on the preference of the players and the number of ships in the game. The players interact with the board by first placing all their ships on the board and afterwards, recording shots and misses during the game.

3.2.2 Game play

The game has two phases:

Type	**Size**
aircraft carrier	5
battleship	4
submarine	3
cruiser	3
destroyer	2

Table 3.1: The type of the ships and their sizes

1. Before starting the game, players place their ships secretly on their grids (Figure 3.1). Ships occupy consecutive squares on the grid, positioned either vertically or horizontally. The two players have equal number of ships of the same type, this number is normally five but can vary (Table 3.1).

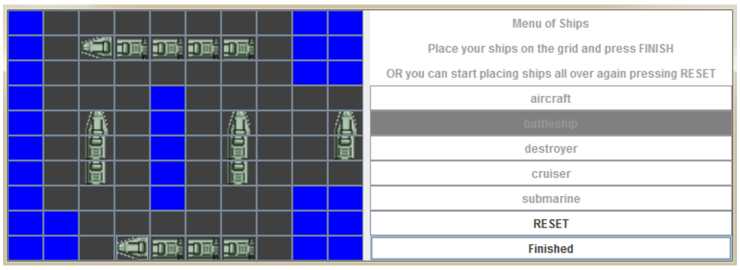


Figure 3.1: The left window shows the grid of the players where the ships can be placed. The right window displays the possible ships that can be placed by the player on the grid.

```
Agent: it's a MISS
Player: it's a MISS
Agent: it's a MISS
Player: it's a HIT
Agent: it's a MISS
```

Figure 3.2: The previously placed ships can be seen on the left window. The player can see all the attempts of his opponent on this window. The middle window shows the game state, and also gives the option for players to choose whether toss or not to toss the die (see more information on this below). The tries of the player are displayed on the right window.

2. After having all the ships placed, the game starts. In each turn, one of the players announces a target square in the opponent's grid (Figure 3.2). If a ship occupies the given square, the opponent announces that it is a hit, otherwise a miss. If all parts of the ship have been hit then the ship is sunk, which has to be announced as well. The game continues until one player wins by sinking all of his opponent's ships.

3.2.3 Die

Our game has a die feature which slightly changes the game play. In the beginning of each turn, players have the choice of tossing the die. If players toss a high number such as 4, 5, 6 then they will be rewarded by a new turn. In case of tossing a low number such as 1, 2, 3 players lose the chance of shooting in the given turn (Table 3.2). The die feature has been added because it is considered to be an interesting additional feature. It might bring about more strategic thinking in the game play. By strategic thinking, we mean that our agent can make a decision whether toss or not to toss the die.

We also need to define what we mean by turns and rounds (Figure 3.3 illustrates a turn). After every turn, the environment is evaluated and an emotion is triggered. If both players have taken their turns then the round is over, which is followed by the next round. Each player, however, might have multiple turns

Number on die	1	2	3	4	5	6
New turn	No	No	No	Yes	Yes	Yes

Table 3.2: The die values and whether they provide extra turn

in a round due to the implementation of the die feature.

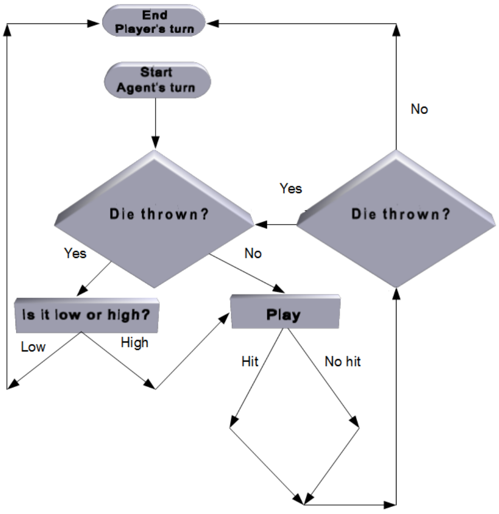


Figure 3.3: Flowchart of a turn

3.2.4 Strategy

Mathematics provides the best strategy: one should start shooting every fifth cell. After 20 turns, this would result in a 100% chance to hit the aircraft carrier (size 5), 80% chance for the battleship (size 4), 60% chance of hitting the cruiser or the submarine (size 3 each), and 40% chance to hit the destroyer (size 2). If there is a hit, the player should sink that given ship trying to find its location by hitting all directions around his first hit. If any of the ships has been sunk, the pattern can be adjusted to the size of the remaining ships (Figure 3.4). However, this feature has not been implemented in the game yet.

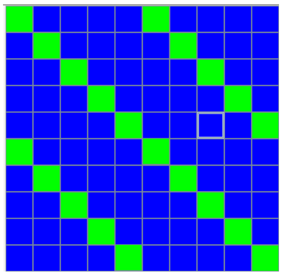


Figure 3.4: The best strategy for finding aircraft carrier (size 5)

People normally toss die when they have fewer ships on the board than their opponents in order to minimize the chance of losing turns. In the game, every time if a player has fewer ships than his opponent then the player will toss the die until they have equal number of ships.

Chapter 4

Translating Scherer's Component Process Model into a computation model

In this chapter, we give a short overview about the appraisal variables, which are the most important parts of the appraisal theories. They help to characterize the interpretation of the environment from an individual's point of view. Based on this personal interpretation, emotions will be elicited. We also discuss other important characteristics of the theory such as the order and the values of the appraisal variables. Furthermore, we suggest including different emotion types for our model. Finally, we propose a mapping from appraisal variables to emotions (computational model).

4.1 Stimulus evaluation checks (Appraisal variables)

There are 4 main phases defined in Scherer's work: relevance detection, implication assessment, coping potential and normative significance. Each of these phases consists of a diverse number of appraisal variables, which may need to be qualitatively and quantitatively simplified. This simplification is inevitable in some cases because Scherer's research is too detailed on appraisal variables for our purposes.

4.1.1 Relevance detection

Scherer presumes that there is an initial phase in the appraisal process, in which the relevance of an event is determined. In other words, if an event is significant enough to require attention, further information processing, or adaptive reaction of a person, then an emotion should be elicited. According to Scherer, the relevance detection phase consists of three appraisal variables:

- Novelty: this appraisal variable can be even further divided into 3 phases. Firstly, it is checked whether any *sudden* stimulus needs to obtain attention (sensory motor level). Secondly, the degree of *familiarity* with the object or event is also examined (schematic level). Thirdly, the novelty evaluation depends on complex estimates of probability and *predictability* of the occurrence of the event (conceptual level).

- Intrinsic pleasantness: an event may result in pleasantness or unpleasantness depending on the personal preference of the individual.
- Goal relevance: an event is relevant if it significantly affects goals. If an event is not significant enough then it will not be appraised and, consequently, no emotion will be triggered.

4.1.2 Implication assessment

This phase determines the extent to which a situation endangers or satisfies one's needs and goals.

- Causal attribution (agent/motive): this appraisal variable identifies the causes of an event. It is important to do so if we want to distinguish emotions such as rage and shame. A person normally feels rage about the consequences of an event that has been done by another person and in case of shame, the person himself has done something wrong.
- Outcome probability: this variable assesses the probability of an event.
- Discrepancy from expectation: this variable determines whether the situation created by an event is consistent or discrepant with the individual's expectation.
- Goal/need conduciveness: this variable is responsible for determining the valence of the emotion. If there is progress towards attaining the person's goal then conduciveness will increase, and eventually result in an emotion with positive valence. If there is no progress then negative emotion will be triggered.
- Urgency: the more important or endangered the goals that are affected by an event, then the more urgent the action becomes for the individual to deal with the situation.

4.1.3 Coping potential

Coping potential checks whether the individual can deal with the consequences of an event. It also determines which type of responses to an event are available and which consequences will affect the individual under each option.

- Control: this variable investigates whether the individual can influence the outcome of an event or not.
- Power: if the control variable refers to possible control over a stimulus, then power variable will determine the extent of the controllability.
- Adjustment: adjust variable is the measure of necessity to adjust or adapt to the consequences of an event.

4.1.4 Normative significance evaluation

This phase is relevant for socially organized species, that are able to evaluate self-concepts, norms and values. During this phase, the evaluation takes place of one's personal self-deal, moral and internal standards, as well as the compatibility of actions with the perceived norms or demands of the majority of a group.

- Internal standards: one's personal self-deal, moral and internal standards are evaluated in this variable.
- External standards: the external standards variable evaluates to what extent an action is compatible with the perceived norms or demands of the majority of a group.

4.1.5 Appraisal variables in our computational model

Scherer has altogether 16 appraisal variables defined in his work.

Appraisal variables = {*Suddenness, Familiarity, Predictability, Intrinsic pleasantness, Goal relevance, Cause agent, Cause motive, Discrepancy from expectation, Outcome probability, Goal conduciveness, Urgency, Control, Power, Adjustment, Internal standards, External standards* }

The number and the type of the appraisal variables depend on the given research (Table 4.1). Implementing all the 16 appraisal variables is not always necessary because they have been created to cover any real life situation in the psychological research. However, games do not necessarily need them all.

Interestingly, according to Scherer, the number of the appraisal variables can be highly reduced and mapped to a three dimensional theory. Valence, activation and power dimensions can be linked to his appraisal variables (valence = goal conduciveness, activation = urgency, power = coping potential). In our model, we can reduce the number of appraisal variables and change their meaning in a manner, so that they fit into our game. In this section, we discuss the appraisal variables that will be removed or simplified. The rest of them will be introduced in the *"Defining appraisal variables "* chapter.

Research	**Emotions**
Broekens et al.	Suddenness, Familiarity, Intrinsic pleasantness, Goal relevance, Goal conduciveness, Urgency, Control, Power
Marinier et al.	Suddenness, Unpredictability, Intrinsic pleasantness, Goal relevance, Causal agent, Causal motive, Outcome probability, Discrepancy from expectation, Goal conduciveness, Control, Power
Courgeon et al.	Expectedness, Unpleasantness, Goal hindrance, External causation, Coping potential, Immorality, Self consistency

Table 4.1: The name of the appraisal variables, which are used in related work

Relevance detection

In the seminal work of Scherer's, the *novelty* variable was called *unexpectedness* [30]. In his later work, he separated this into three levels (*suddenness*, *familiarity* and *predictability*)[31]. *Expectedness/unexpectedness* appraisal variables are already used in practical works [7, 6]. Defining all the three levels of novelty would require a lot of knowledge, and thus, result in a too complex model. In our opinion, the *unexpectedness* variable is sufficient to describe the novelty of events.

Intrinsic pleasantness is considered separate from, and prior to, the *goal conduciveness* appraisal variable. *Goal conduciveness* determines whether an emotion is positive or negative. One could argue that *intrinsic pleasantness* is not necessary because *goal conduciveness* will determine the valence of the stimulus anyway. Moreover, in most cases the values of these two correspond, but we must keep in mind that there may be cases when an event is possibly pleasant but obstructive for goals, or unpleasant but somehow contributes to achieving a goal. For example, if a person keeps eating chocolate, that activity results in pleasure. However, this will not contribute to his goal to lose weight. In a computer game, this could be translated that a player keeps destroying all objects in a room although it would be clear that it will not help him to achieve his goals. In our work, the valence of the triggered emotions will be entirely based on the *goal conduciveness* variable, omitting the *intrinsic pleasantness* variable.

Implication assessment

Causal attribution is separated into two variables in Scherer's work: agent and motive. However, we have decided to merge the two and postulate, that our emotion triggering model is only capable of recognizing one cause at a time. This way, we eliminate the possibility of multiple causes of events.

The *urgency* variable determines whether players need to act quickly after an event occurs. This variable would be quite important in the case of continuous time but our research uses discrete time. This means that there is an equal time portion between turns, so the *urgency* variable would not have any role in our system.

Normative significance evaluation

Understanding social situations and interpreting them in the right manner is quite complicated. For example, in a poker scenario, when a player receives bad cards then he may attempt to hide his emotions, and bluff. This requires a deep understanding of the environment and an entirely new research topic could be based on it. Thus, *internal* and *external standards* appraisal variables are excluded from our work.

Appraisal variables in our work = {*Unexpectedness, Goal relevance, Causal attribution, Outcome probability, Discrepancy from expectation, Goal conduciveness, Control, Power, Adjustment* }

4.2 Order

According to Scherer, appraisal variables are being processed in sequence following a fixed order [31]. The idea of sequential processing is economically beneficial: if a stimulus does not require any attention (stimulus is not goal relevant) then there will be no need to further evaluate the event.

Gratch & Marsella challenge the assumption of sequential processing, particularly the fixed order idea. They think that appraisal variables should be evaluated in parallel. Marinier et al. claim that appraisal should have at least a partial ordering. In their work, the appraisal variables are adjusted to the PEACTIDM cycle.

In our research, the variables are evaluated in a fixed order. The order does not correspond to Scherer's model and the reasons for that will be discussed in the "*Proposed mapping*" section in detail. Here, we provide a short outline of the order.

First, the relevance (*goal relevance*) of an event is measured. If the event is not significant enough then there is no further computation. If it is significant enough then the valence (*goal conduciveness*) of the event is checked. This might save a lot of calculation time, because emotions with opposite valence do not have to be taken into account anymore. Finally, the rest of the appraisal variables will be taken into account, except the *causal attribution* variable, which is used to refine the triggered emotion in the end of the computation.

4.3 Emotions

Numerous studies have been carried out in order to find a comprehensive list of all emotion terms. For example, one of the studies has reported more than 500 English terms [30]. These studies, however, did not take into account that there are many synonyms, which describe the same emotions slightly differently. For example, rage and anger are synonyms. People would be fine with hearing any of the two as an appropriate emotion to describe a person's emotion that is in a frustrating situation. The salient advantage of using appraisal variables is that they are capable of pointing out these subtle differences between emotions.

As opposed to discrete emotion theories, Scherer does not believe in a limited number of emotions that mix or blend in order to produce different emotional states. He believes in a continuous emotion space with dimensions that are equal to the number of appraisal variables. Nonetheless, he admits that there are some major recurring patterns in the continuous emotion space, which yield emotions. In his research, he specifies 14 emotions and their eliciting conditions as well.

Emotions = {*Enjoyment / Happiness, Elation/ Joy, Displeasure/ Disgust, Contempt/ Scorn, Sadness/ Dejection, Despair, Anxiety/ Worry, Fear, Irritation/ Cold Anger, Rage/ Hot Anger, Boredom/ Indifference, Shame, Guilt, Pride*}

4.3.1 Emotions in our research

The game creators may decide themselves which emotion types they would like to implement in their games. In our work, we select emotions for the Battleship

Research	**Emotions**
Paleari et al.	Happiness, Disgust, Contempt, Sadness, Pride, Fear, Anger, Indifference and Shame
Malatesta et al.	Happiness, Disgust, Contempt, Sadness, Pride, Fear, Anger, Indifference and Shame
Marinier et al.	All
Courgeon et al.	Joy, Sadness, Anger and Guilt
Broekens et al.	Unknown

Table 4.2: The type of emotions in the related work

game based on the following criteria:

- easily distinguishable from other emotions: there are a few emotions that are closely related, so one can barely differentiate between them (e.g. hot anger and cold anger or worry and fear). Thus, we aim at finding emotions that are easily distinguishable and exclude all those that are alike in terms of appraisal variable values.
- suggested in research that use Scherer's theory: the literature suggests the use of various emotions. We take into account the reasons why certain emotions have been chosen by researchers (Table 4.2)

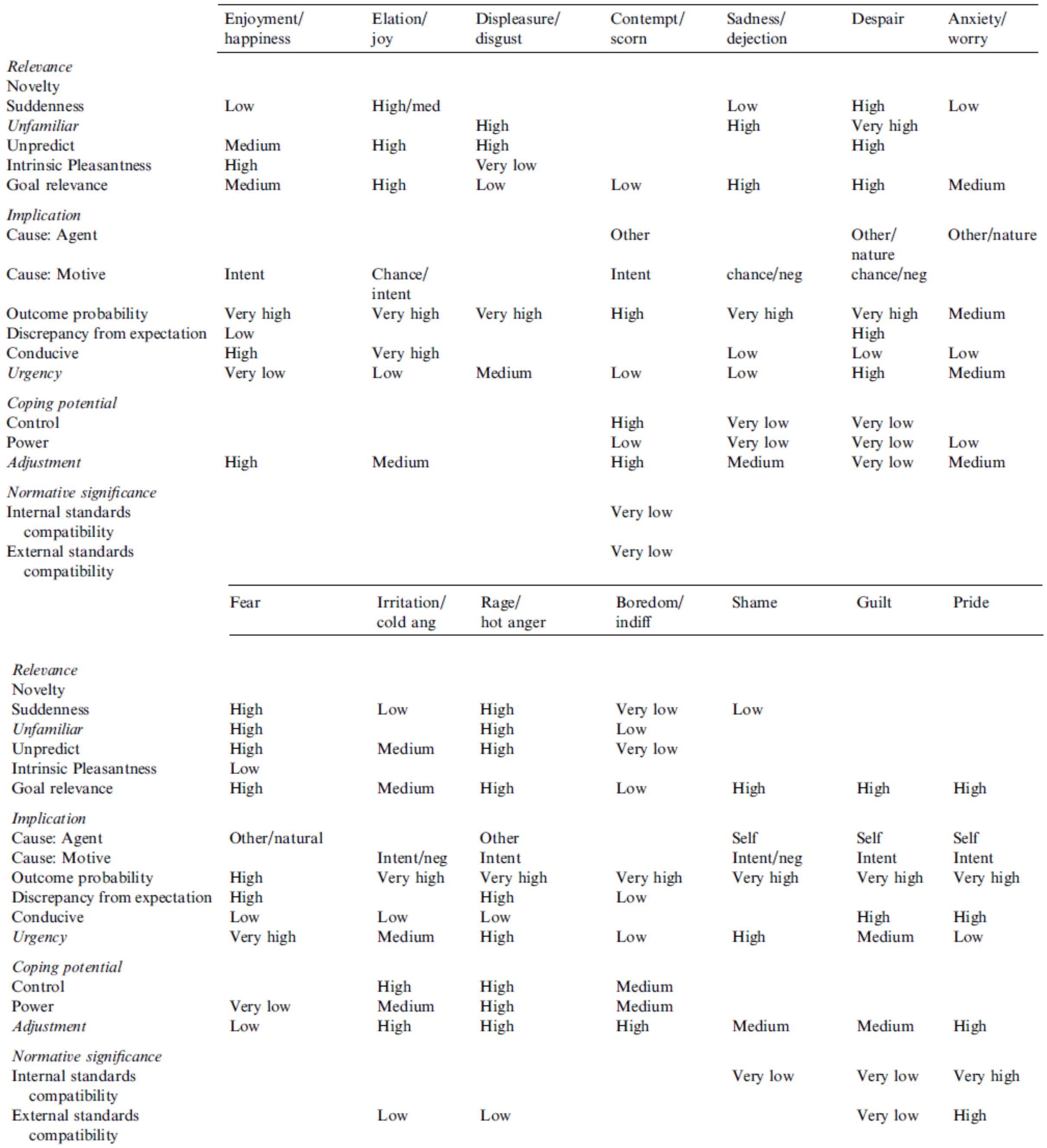
Examining our criteria and the proposed emotions by Scherer, we have come to the conclusion to implement six emotions: happiness, rage, fear, shame, pride and sadness. It can be clearly seen that happiness-sadness and shame-pride are opposite pairings, which are easily distinguishable based on their valence. The number of emotions implemented can be easily extended as we will describe it in the *"Implementation"* chapter.

Emotions in our model = {*Happiness, Sadness, Fear , Rage, Pride, Shame*}

4.4 Appraisal variable values

Figure 4.1 specifies the mapping between appraisal variables and emotions. Appraisal variable values may carry more information than a simple binary "*yes*" or "*no*". There are various labels assigned to each appraisal variable such as "*very low*", "*low*", "*medium*", "*high*", "*very high*" or "*open*". The term "*open*" may sound confusing but it is used if various labels of a particular appraisal variable are compatible or completely irrelevant. Let us describe an emotion in the same manner as Scherer specifies it. For example, the following conditions need to be fulfilled to trigger rage:

Rage: A highly unexpected and goal relevant stimulus occurs in the game, caused by either the player's team mate or opponent. The stimulus is negatively diverse from the player's expectation and does not help him to achieve his goal at all. Fortunately, the consequences can be kept under control to a high extent but some adjustment will be needed.



	Enjoyment/ happiness	Elation/ joy	Displeasure/ disgust	Contempt/ scorn	Sadness/ dejection	Despair	Anxiety/ worry
Relevance							
Novelty							
Suddenness	Low	High/med			Low	High	Low
Unfamiliar			High		High	Very high	
Unpredict	Medium	High	High			High	
Intrinsic Pleasantness	High		Very low				
Goal relevance	Medium	High	Low	Low	High	High	Medium
Implication							
Cause: Agent				Other		Other/ nature	Other/nature
Cause: Motive	Intent	Chance/ intent		Intent	chance/neg	chance/neg	
Outcome probability	Very high	Very high	Very high	High	Very high	Very high	Medium
Discrepancy from expectation	Low					High	
Conducive	High	Very high			Low	Low	Low
Urgency	Very low	Low	Medium	Low	Low	High	Medium
Coping potential							
Control				High	Very low	Very low	
Power				Low	Very low	Very low	Low
Adjustment	High	Medium		High	Medium	Very low	Medium
Normative significance							
Internal standards compatibility				Very low			
External standards compatibility				Very low			

	Fear	Irritation/ cold ang	Rage/ hot anger	Boredom/ indiff	Shame	Guilt	Pride
Relevance							
Novelty							
Suddenness	High	Low	High	Very low	Low		
Unfamiliar	High		High	Low			
Unpredict	High	Medium	High	Very low			
Intrinsic Pleasantness	Low						
Goal relevance	High	Medium	High	Low	High	High	High
Implication							
Cause: Agent	Other/natural		Other		Self	Self	Self
Cause: Motive		Intent/neg	Intent		Intent/neg	Intent	Intent
Outcome probability	High	Very high	Very high	Very high	Very high	Very high	Very high
Discrepancy from expectation	High		High	Low			
Conducive	Low	Low	Low			High	High
Urgency	Very high	Medium	High	Low	High	Medium	Low
Coping potential							
Control		High	High	Medium			
Power	Very low	Medium	High	Medium			
Adjustment	Low	High	High	High	Medium	Medium	High
Normative significance							
Internal standards compatibility					Very low	Very low	Very high
External standards compatibility		Low	Low			Very low	High

Figure 4.1: A mapping from SECs to modal emotions (from [31]). Abbreviations: All values are allowed in open cells. Unfamiliar = unfamiliarity, unpredict = unpredictable, conducive = conduciveness, med = medium, intent = intentional, neg = negligence, ang = anger, indiff = indifference.

4.5 Existing approaches to mapping

Scherer defines emotions as the combination of different values of the appraisal variables. The problem is that obtaining values from a game environment will not necessarily give us these precise formations of the values that are specified in Scherer's work for the emotions. Unfortunately, Scherer does not share any information on how to handle this particular problem. However, existing literature has already attempted to map the values of the appraisal variables to emotions.

For example, Gratch & Marsella suggest that the values of the appraisal variables should be translated into two separate categories: categorical and numerical [13]. In this separation, the categorical values are only used for determining the cause of events. The rest of the appraisal variables can be translated into numerical values in the range of 0 and 1, in which the 0 end is less intense than the 1 end. Every time an event occurs, numerical values are assigned to appraisal variables and based on that, an emotion is calculated. The EMA framework illustrates six emotions such as hope, joy, fear, distress, anger and guilt. Thus, they think that only a few appraisal variables (desirability, likelihood and causal attribution) are sufficient for emotion triggering. All the emotions are defined from these appraisal variables, measuring the differences between their numerical values.

Marinier et al. have slightly modified the range for the appraisal variables [19]. They assume that numerical values should use an extended range into the minus direction as well. It would mean that values can vary from -1 to 1. Their supporting idea is that the "*low*" value could be just as intense as the "*high*" value in the case of using numerical values. For example, *intrinsic pleasantness* may have the value of -1, indicating high unpleasantness, whereas 1 would be highly pleasant. They believe that only a few appraisal variables need the range between -1 and 1, providing no valid explanation why the rest should not take values from this range. Their mapping function is based on measuring the Manhattan distance between appraisal values derived from an environment by an agent and all the emotions specified in Scherer's table. They expect the emotion with the smallest value to be triggered. They give an example how they map the appraisal variables and their values to emotions. However, in this example, they apply only extreme values such as ("*low*" and "*high*"), disregarding the rest of the labels.

In the work of Courgeon et al., Gaussian curves are defined to enable appraisal variables evaluation on a continuous scale [7]. This provides a less restricted computation. Each of the labels - that are used for emotion mapping in Scherer's CPM - is assigned to a Gaussian curve. When an event occurs, its relevance to all appraisal variables is evaluated. For each emotion, the relevance is sequentially multiplied with the Gaussian of the emotion.

The problem is that these works do not reveal much information on how the appraisal variable values are defined, which will be the main subject of the next chapter. For example, if a certain event occurs why precisely should "*low*", "*high*" or any numerical value be assigned to an appraisal variable. However, the literature provides information on how to map appraisal variables to emotions. In the next section, following these ideas, we will develop a computational model.

4.6 Mapping from appraisal variables to emotions

Using categorical labels (*"very low", "low", etc.*) means that there will be five possible labels that can be assigned to appraisal variables. In addition, the "*open*" label may be also assigned, implying that the particular appraisal variable may take any value.

There are two appraisal variables, namely *unexpectedness* and *causal attribution*, to which we need to assign values by ourselves, because they are not present in Scherer's mapping table, due to our previously discussed simplifications. As we mentioned before, *unexpectedness* is already discussed in Scherer's earlier paper [30], in which he specifies its values using a binary scale. We apply the same values, which means that the *unexpectedness* variable will take only "*high*","*low*" and "*open*" labels in our work. Furthermore, there are emotions in Scherer's work, to which he does not specify any value of *unexpectedness*. To overcome this problem, we just simply assign the "*open*" value to these emotions.

Merging *cause agent* and *motive* into the *causal attribution* appraisal variable raises the problem that we need to manually find values for our new appraisal variable. Our work implements labels such as *chance*, *other* and *self*. Table 4.3 illustrates the modified values of the appraisal variables of the previously selected six emotions.

	Happiness	**Pride**	**Sadness**	**Fear**	**Rage**	**Shame**
Unexpected.	Low	?	?	High	?	?
Goal r.	Medium	High	High	High	High	High
Causal a.	Chance	Self	Chance	Other	Other	Self
Outcome p.	V. High	V. High	V. High	High	V. High	V. High
Discrepancy	Low	?	?	High	High	?
Goal c.	High	High	Low	Low	Low	?
Control	?	?	V. Low	?	High	?
Power	?	?	V. Low	V. Low	High	?
Adjustment	High	High	Medium	Low	High	Medium

Table 4.3: The values of the appraisal variables of the selected emotions

After discussing the values of the new appraisal variables, we describe step-by-step how we can trigger emotions. In the case of structural appraisals (i.e. OCC model), computation is easier because there are only two possible labels for each appraisal variable. In Scherer's work, there are altogether five values that can be assigned to appraisal variables and the "*open*" label. Similarly to Marinier et al., we would like to measure the similarity between appraisal variable values of the game and of the emotions specified by Scherer's mapping table. In order to do so, we first convert categorical labels into numerical values (Table 4.4).

Using a categorical scale instead of continuous gives us the advantage that the distance is always the same between any two labels next to each other. For example, the difference is 2 between "medium" and "very high" values. Thus, we can apply the Manhattan distance function to find out the similarity. This

Labels	**Numerical values**
Open	no calculation
Very Low	-2
Low	-1
Medium	0
High	1
Very High	2

Table 4.4: Conversion from labels to numerical values

function computes the distance that is travelled from one data point to the other. For example, the Manhattan distance between two items ($X = (x_1 \ldots x_n)$ and $Y = (y_1 \ldots y_n)$) is the sum of the differences of their corresponding components.

$$distance = \sum_{i=1}^{n} \|x_i - y_i\|$$

We need to slightly modify this formula in order to fit in our research:

$$emotion_i = \sum_{\substack{i=n \\ j=m}}^{\substack{n \\ m}} \|scherer_{ij} - game_j\|$$

where:

- i is an emotion label (in Battleship: happiness, sadness, fear, rage , shame, pride).
- j is an appraisal variable (in Battleship: unexpectedness, goal relevance, causal attribution, outcome probability, discrepancy from expectation, goal conduciveness, control, power, adjustment).
- n is the number of emotion labels (in Battleship: it is 6).
- m is the number of appraisal variables (in Battleship: it is 9).
- $emotion_i$ is the Manhattan distance of the i^{th} emotion.
- $scherer_{ij}$ is value specified in Scherer's mapping table for the j^{th} appraisal variable of the i^{th} emotion.
- $game_j$ is the actual value for the j^{th} appraisal variable derived from the game.

Computing distances in this manner raises issues. It is more likely that emotions with fewer labels are triggered because there is a smaller chance that their values differ. In order to avoid this, we normalize the Manhattan distance by dividing it by the number of the appraisal variables (excluding the appraisal variables with the "*open*" label).

Furthermore, there may be appraisal variable values that are not derived from the game but present in Scherer's mapping. We have to make sure that all the generated appraisal variables in the game will have values assigned, every time when an event is evaluated. The emotion with the smallest normalized Manhattan distance will be the potential candidate to be trigged.

We call it potential because there are three other factors that we still need to take into account to compute the similarity. Firstly, the event must be relevant enough in order to start the computation. This step saves up a lot of calculation time if the event is not relevant.

Moreover, according to our current computation, it could happen that a positive emotion will be triggered although the *goal conduciveness* variable has a negative value. Imagine a game scenario, when the game character is happy despite losing the game. This would be highly inappropriate and such a flaw could ruin completely the credibility of the emotions. Thus, secondly, we need to resolve this issue by separating positive and negative emotions depending on the value of the *goal conduciveness* variable. Our model will be restricted to two labels: ”*low*” and ”*high*” (Table 4.5). In other words, an event can be negative or positive, which implies that there will be no further separation between values. However, not all the emotions specify labels for the *goal conduciveness* variable because there are emotions that do not belong exclusively to any of the previous two categories. For example, shame is neither a positive nor a negative emotion. A person can be ashamed when he has done something wrong for another person but the result of his action would be beneficial to achieving his own goal. Luckily, these particular situations are excluded from our model since we do not take into account the *external* and the *internal standards* variables. Thus, only a slight modification is required regarding Scherer's mapping. The *goal conduciveness* variable of shame should be modified from ”*open*” to ”*low*” value.

	Happiness	**Pride**	**Sadness**	**Fear**	**Rage**	**Shame**
Goal c.	High	High	Low	Low	Low	Low
Valence	Positive	Positive	Negative	Negative	Negative	Negative

Table 4.5: The valence of the emotions is given by the *goal conduciveness* variable

The third factor is the *causal attribution* variable. The cause of an event may change the potential emotion. For example, in the case of a positive event, happiness would have the smallest value of *0.8* and pride would have *0.9*. However, the cause of the event is the agent itself, thus pride would be a more appropriate emotion than happiness. Thus, a range should be defined for refinement, which helps to choose which type of emotion to trigger if emotions have equal or extremely close distance values (Table 4.6) but their causes are diverse. Figure 4.2 illustrates an overview of our computational model.

	Happiness	**Pride**	**Sadness**	**Fear**	**Rage**	**Shame**
Causal a.	Chance	Self	Chance	Other	Other	Self

Table 4.6: The *causal attribution* appraisal variable of the emotions.

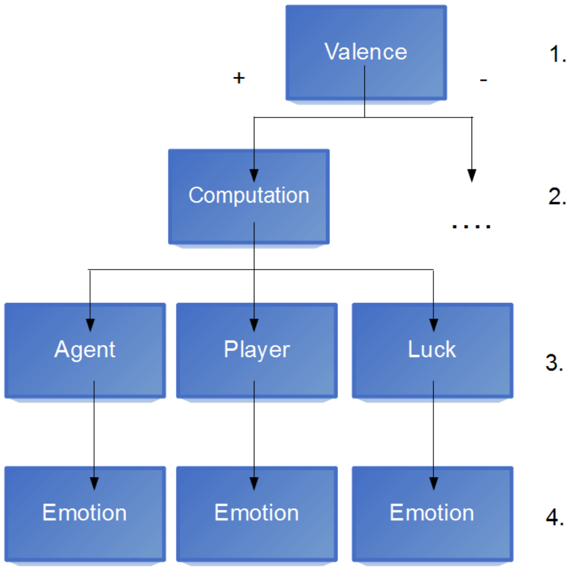


Figure 4.2: Before computation starts, *goal relevance* is checked. If a goal is significantly affected then computation starts: 1. Valence is determined. Knowing the valence automatically excludes the emotions that have opposite valences. 2. The Manhattan distance is computed. 3. The causal attribution phase refines the potential emotion. 4. An emotion is triggered

Example of emotion mapping in Battleship

Here, we give an example that shows how this previously introduced emotion triggering works in a real game. We first describe a game scenario, and based on that situation, values will be assigned to the appraisal variables. The values are assigned manually to this game scenario as our educated guesses. In the next chapter, these values are determined by the NPC's beliefs, goals and intentions. Imagine the following game scenario from the NPC's aspect.

Battleship game scenario: The NPC is very certain about where its opponent's ship is located, because it has hit the ship already two times. In this turn, it hits the ship again, and also sinks it. This way, it has one more ship remaining than its opponent.

NPC's interpretation: The event is *expected* because the agent already knew the location of the ship, so the *probability* was quite high that the agent will hit its opponent's ship again. Since the ship is sunk, the *goal relevance* becomes high as well. The event was *caused* by the agent itself because it selected the correct grid to hit the ship. This means that the event was not *discrepant from the expectation* at all. Furthermore, its action contributes to achieving its goal, so *goal conduciveness* of the event is high. The agent has a lot of *control* as well as *power* because it has a high probability of winning the game. The NPC's actions need only a little *adjustment*.

The event has positive valence due to its high goal conduciveness. This means that only emotions with positive or unspecified valence need to be taken into account such as pride and happiness.

1. Determining whether the event is significant enough.
2. Interpreting the game situation in terms of the appraisal variables (Table 4.7).
3. Converting the appraisal variable values of the potential emotions excluding emotions with opposite valence (Table 4.8).
4. Measuring the Manhattan distance between the appraisal variable values of the potential emotions and the game (Table 4.9).
5. Finally, the *causal attribution* variable is taken into account as well. In this particular example, the causal attribution range is set to *0.2* (Table 4.10).
6. Pride will be triggered after the refinement phase.

Appraisal v.	**Event**
Unexpectedness	Low
Goal relevance	High
Causal a.	Self
Outcome p.	High
Discrepancy	Low
Goal c.	High
Control	High
Power	High
Adjustment	Low

Table 4.7: The appraisal variable values derived from the game

Appraisal v.	**Happiness**	**Pride**
Unexpectedness	$Low \rightarrow -1$	?
Goal relevance	$Medium \rightarrow 0$	$High \rightarrow 1$
Outcome p.	$V.High \rightarrow 2$	$V.High \rightarrow 2$
Discrepancy	$Low \rightarrow -1$	?
Control	?	?
Power	?	?
Adjustment	$High \rightarrow 1$	$High \rightarrow 1$

Table 4.8: The values of the appraisal variables and their numerical values

Appraisal v.	**Happiness**	**Pride**	**Game event**
Unexpectedness	$-1 \rightarrow 0$	?	$Low \rightarrow -1$
Goal relevance	$0 \rightarrow 1$	$1 \rightarrow 0$	$High \rightarrow 1$
Outcome p.	$2 \rightarrow 1$	$2 \rightarrow 1$	$High \rightarrow 1$
Discrepancy	$-1 \rightarrow 0$	?	$Low \rightarrow -1$
Control	?	?	$High \rightarrow$1
Power	?	?	$High \rightarrow$1
Adjustment	$1 \rightarrow 2$	$1 \rightarrow 2$	$Low \rightarrow$ -1
Manhattan distance	4	3	
Number of values	5	3	
Normalized Manhattan	0.8	1	

Table 4.9: The table illustrates the converted appraisal values and their distances. Below, there are the values of the Manhattan distance, the number of the appraisal variables and the normalized Manhattan distance

Appraisal v.	**Happiness**	**Pride**	**Game event**
Manhattan distance	4	3	
Number of values	5	3	
Normalized Manhattan	0.8	1	
Causal a.	Chance	Self	Self (range = 0.8 + 0.2)
Potential	Yes	Yes	
Triggered	No	Yes	

Table 4.10: The refinement of the computation by including the *causal attribution* appraisal variable

Chapter 5

Defining appraisal variables

When people are asked to consider a situation, they tend to explain their actions in terms of their intentions, which in turn are explained in terms of their goals and beliefs. For example, Bob is an agent who believes that he does not have any milk at home, but he desires to drink some. Thus, going to the supermarket and buying some milk becomes Bob's plan, which he commits to (intention). The mental attitudes such as beliefs, desires and intentions represent the person's cognitive representation of the world at any given time. Our goal is to link these mental attitudes to the appraisal variables - which are used to characterize the subjective interpretation of the events.

In the previous chapter, we have already discussed how to map the values of the appraisal variables to emotions (computational model). Thus, in this chapter, we focus on defining the meaning of the appraisal variables, and give detailed information on the way, in which they obtain values. Firstly, we introduce the BDI theory in a nutshell. Next, we present the notation. A few new definitions will be needed because beliefs, desires and intentions are not always enough to describe all the appraisal variables. Finally, we define all the appraisal variables. We have to stress that our work is not oriented towards any formal logical framework. This means that all the presented formula in this chapter are semi-formal, because no semantics is given, but logical connectives and operators will be used with their normal interpretation.

5.1 BDI theory

The original model of BDI was developed in the study called Intention, Plans and Practical Reason by the philosopher Michael Bratman [5]. He claims that an agent is characterised by its beliefs, desires (goals) and intentions (the agent will do what it believes will help its goal, given its beliefs about the world).

Beliefs can be acquired in many ways, such as perception, contemplation or communication. Basically, beliefs are facts that represent what an agent believes about its environment and are appropriately updated every time any change occurs. It is important to mention that the agent's beliefs are not necessarily true.

Desires can be viewed as goals or as some desired end states. They represent the motivational state of the BDI system. Desires carry information about objectives that need to be accomplished. Agents may have multiple desires, which may possibly be in conflict. The literature often refers to desires as goals. Henceforth, goal and desire terms will be used interchangeably in this thesis.

Intentions may refer to an agent's commitments to its desires, and its commitment to the plans selected to achieve those goals. They cannot conflict with each other. Firstly, it may seem that beliefs and desires would be enough to determine which action to select at any point in time. However, without the intention component, the agent's behaviour would become inefficient. For example, in the case of multiple desires, the agent would be stuck. It would not know which goal to execute first. Thus, the agents must have intentions to construct plans and commit to them.

5.2 Existing literature

Some existing research has attempted to provide a partial or complete formalization of the OCC model in logic [1, 33, 22]. Our goal is entirely different but we can still benefit from these works. For example, Steunebrink formalizes the eliciting conditions of emotion in three stages in his work [33]. They are separated because, this way, different levels of commitment to formalism is provided. The first level is not oriented towards any logical framework, and thus, presents semi-formal rules, since no semantics is given, while the remaining two levels commit to formalism. This way, the first level gives us an idea on how to provide semi-formal definitions for the appraisal variables, which is our goal in this chapter. However, the OCC model does not share much similarity with Scherer's work, mainly because the eliciting conditions are absolutely different. Scherer defines five values and the possibility that a given appraisal variable is not included in the emotion elicitation process, while the OCC model follows a simple binary structure. In conclusion, the existing research is not fully relevant to our work. Therefore, we do not describe them in details.

5.3 Notation

Our research investigates how an emotion is elicited from the aspect of an individual. Henceforth, we call this individual an agent, who is denoted i. This agent is capable of taking an action from the set of actions that is denoted $\pi = \{\pi_1, \pi_2, \ldots, \pi_f\}$, where π_j, $1 \leq j \leq f$, represents a specific distinct action. Furthermore, E is a set of events that may occur in the game, E = $\{e_1, e_2, \ldots, e_l\}$, where e_k, $1 \leq k \leq l$, takes a value based on the result of k^{th} action, which has been chosen immediately prior.

In our understanding, the main phases of the appraisal process can be divided into two groups. The *relevance detection* and the *implication assessment* phases deal with the consequences of current events, while the *coping potential* phase checks whether the agent will be able to cope with the consequences of an event in the future. Therefore, in order to express future, we need to define time as well: the game time is described in terms of a series of timepoints $t_1, t_2, \ldots t_m$, where t_n, $1 \leq n \leq m$ represents the moment when agent i has just taken his n^{th} turn.

Agent i can believe that something is true or false in its world. For example,

$\mathbf{Bel}_i$ (X) is read as "*agent i believes that X is currently true*". Beliefs are always relative to something, so X represents "*something*" here. If "*agent i believes that X is not true*" then the previous belief is modified by a negation sign: $\mathbf{Bel}_i$ $(\neg X)$.

We should be more clear on what type of values X can take. In the OCC model, X could be an event, an action or an object. However, Scherer uses only the stimulus term without further specifying the reason why the emotion is triggered. In our model, we will always assume that X is an event that has just occurred at time t_n. In our opinion, there is no need to define actions because all the taken actions lead to events, while objects will be excluded from our model. X is the direct consequence of a change or an action in the game environment. For example, in our Battleship game, in the *10th* turn, if agent i hits a ship then that would be written as: at time t_{10}, $\mathbf{Bel}_i$ (*hit(ship)*).

After clarifying the meaning of X, we can define desires (goals) as well. "*Agent i desires X to be true*", which is written as: $\mathbf{Des}_i$ (X). However, if "*the agent does not desire X to be true*" then: $\mathbf{Des}_i$ $(\neg X)$.

There are also several other factors that we need to define in order to provide values for all the appraisal variables. To be more precise, the meaning of probability, cause and relevance needs to be also defined. The probability of certain events to occur is needed because the value of the previously introduced appraisal variables - such as *unexpectedness, outcome probability* and *discrepancy from expectation* - is derived from that. The probability of events will be also given as beliefs for the agents. For example, "*agent i believes that event X occurs with the probability y*".

$\mathbf{Bel}_i\mathbf{P}(X) = y$, where $y \in [0, 1]$

Furthermore, the cause of an event must be also known for the agent in order to implement *causal attribution* variable. Thus, cause will be assigned to each event. In our current model, there are three possible causes: self, other and chance. The belief of an agent is slightly modified:

$\mathbf{Bel}_i$ $(cause{:}X) =$ z, where $z \in$ [*self, other, chance*].

To give an example, if agent i itself is the cause of X then it can be rewritten as $\mathbf{Bel}_i$ $(self{:}X)$. If the cause is not specified for X, then we can just simple write $\mathbf{Bel}_i$ (X), without providing the cause.

The goals are normally not equally important, which means that we should define the relevance of the goals (*goal relevance*). The relevance provides the significance of the desires that agent i would like to achieve.

$\mathbf{Rel}(\mathbf{Des}_i(X)) =$ r , where $r \in$ [*Very low, Low, Medium, High, Very high*]

5.4 Appraisal variables

The above notation enables us to define the meaning of the appraisal variables. In our opinion, our definitions, with small modifications can be applied to any kind of game with discrete time. We assume that all the appraisal variables of the first two main phases (relevance detection and implication assessment) take their values at time t_n, which is the actual turn of agent i, directly after it has taken an action. In the coping potential phase, time refers to the future, more precisely, right to the next turn, when the agent will have just taken its action again. This means that time will be modified as t_{n+1}.

5.4.1 Unexpectedness

Unexpectedness is related to probability. If something is unlikely to happen, then one perceives it as unexpected. We believe that depending on the subjective interpretation of probability values, *unexpectedness* can be determined. In our research, "*low*" and "*high*" values may be assigned to this variable. The subjective interpretation is represented by a threshold variable, which can be set manually by game designers (Table 5.1). For example, if the threshold is set to *0.5* then all the events, which have the probability more than 50% to occur will be considered expected, while the rest unexpected. This means that depending on the person's personality, the value of threshold can be different, which would result in appraising events differently. For example, in a Battleship game scenario, a pessimistic person might find hitting a ship unexpected, while an optimistic person would think precisely the opposite.

Unexpectedness = $\{Low, High\}$

Unexpectedness	**Definition**
Low	$\mathbf{P}(X) > threshold$
High	$\mathbf{P}(X) \leq threshold$

Table 5.1: The values of the *unexpectedness* variable

5.4.2 Goal Relevance

To reason about relevance and desirability of a desire (goal), the model must represent preferences over outcomes. This way, the *relevance* value should be assigned to all the goals manually by the game creators, indicating the importance of the goals. There may be events in the game that do not require appraisal because none of the agent's goals has been significantly affected, or their significance is so "*low*". In our research, if multiple goals are affected, then we take the value of the most significant goal (Table 5.2).

Goal relevance	**Name of goal**
Very low	Goal 1
Very high	**Goal 2**
Medium	Goal 3

Table 5.2: "*Very high*" value is assigned to *goal relevance* variable, as that is the highest value of all affected goals

5.4.3 Causal attribution

It is important to identify the cause of an event because it helps to separate emotions such as pride and happiness. According to Scherer's specification, pride is more related to the person's personal choice, whereas happiness is more related to chance, which means that the agent has small control over the happenings. There are three possible causes of an event in the game, and it is only possible for one to be present at any given time: *self*, *other* and *chance*. The *self* value is assigned if the event occurs due to an action taken by the agent. The *other* value is the same as *self* but the action is taken by another agent in the environment. The *chance* value refers to events, on which the agents have a very small influence.

5.4.4 Outcome Probability

Outcome probability variable corresponds with the probability value of a particular event to occur. It may happen that an emotion is triggered due to the occurrence of multiple events. This means that the probability of each of the events needs to be evaluated. In that case, we simply add up all the probability values and divide them by the number of events (n) that are taken into account.

Outcome probability = {*Very low, Low, Medium, High, Very high*}

$$\frac{P(X_1 + \cdots + X_n)}{n}$$

The result of the probability value is translated into a categorical label of the *outcome probability* (Table 5.3).

Outcome probability	**Definition**
Very low	$0 \leq \mathbf{P}(X) \leq 0.2$
Low	$0.2 < \mathbf{P}(X) \leq 0.4$
Medium	$0.4 < \mathbf{P}(X) \leq 0.6$
High	$0.6 < \mathbf{P}(X) \leq 0.8$
Very high	$0.8 < \mathbf{P}(X) \leq 1$

Table 5.3: The values of the *outcome probability* appraisal variable

5.4.5 Discrepancy from expectation

In our research, the *discrepancy from expectation* variable measures the degree of unexpectedness of an event (Table 5.4). In other words, it measures the difference between the threshold and the probability of an event.

Discrepancy from expectation = {*Very low, Low, Medium, High, Very high*}

Discrepancy from expectation	**Definition**
Very low	$0 \leq \| \mathbf{P}(X) - threshold \| \leq 0.2$
Low	$0.2 < \| \mathbf{P}(X) - threshold\| \leq 0.4$
Medium	$0.4 < \| \mathbf{P}(X) - threshold\| \leq 0.6$
High	$0.6 < \| \mathbf{P}(X) - threshold\| \leq 0.8$
Very high	$0.8 < \| \mathbf{P}(X) - threshold\| \leq 1$

Table 5.4: The values of the *discrepancy from expectation* appraisal variable

5.4.6 Goal conduciveness

This is the most important variable that determines whether the emotion has negative or positive valence. This can be determined by viewing goals as achieved or failed. The definition of an achieved goal is if an agent has the belief that something is not true, although it desires it to be (Table 5.5). As previously described, the *goal conduciveness* variable may have "*low*" and "*high*" values.

Goal conduciveness = {*Low, High*}

Goal Conduciveness	**Definition**
Low	$\mathbf{Des}_i(\mathrm{X}) \wedge \mathbf{Bel}_i(\neg X)$
High	$\mathbf{Des}_i(\mathrm{X}) \wedge \mathbf{Bel}_i(X)$

Table 5.5: The values of the *goal conduciveness* appraisal variable

It is possible that an agent achieves one goal but fails another. Thus, we determine the valence by taking into account the goal that has the highest relevance in the case of the presence of multiple goals.

5.4.7 Control and Power

Scherer claims that literature does not always clearly distinguish *control* and *power* variables. In addition, controllability is often used to imply both aspects [31]. In his research, *control* variable refers exclusively to the probability that a person can control the consequences of an event, whereas *power* variable determines the probability that an agent can influence a controllable event. In our research, they are not fully separated and normally both have the same values assigned.

If there is a possibility that the agent can respond to the consequence of certain event then it will be considered controllable. If there is not then the event becomes uncontrollable. This means that *control* variable may take only extreme values such as "*low*" and "*high*" (Table 5.6). The controllable event can be translated as follows: agent i desires X to be true but currently it believes that it is not. However, there is at least one action which can be taken by agent i, which may cause X to be true in the future.

As an alternative definition, the *power* variable can be seen as a measurement of how much an event can be controlled. It is determined based on the number of actions, which the agent can take in a specific situation.

Control and Power = {*Low, High*}

Control and Power	**Definition**
Low	At t_n: $\mathbf{Des}_i(\text{X}) \wedge \mathbf{Bel}_i(\neg X) \wedge (\neg\exists\pi \rightarrow$ at t_{n+1}: $\mathbf{Des}_i(\text{X}) \wedge \mathbf{Bel}_i(X))$
High	At t_n: $\mathbf{Des}_i(\text{X}) \wedge \mathbf{Bel}_i(\neg X) \wedge (\exists\pi \rightarrow$ at t_{n+1}: $\mathbf{Des}_i(\text{X}) \wedge \mathbf{Bel}_i(X))$

Table 5.6: The values of the *control* appraisal variable

5.4.8 Adjustment

The *adjustment* variable measures how much the agent needs to adjust after an event has just occurred. In our research, we translate *adjustment* as the measurement of the probability that an agent can win the game, or achieve a particularly important goal. The probability is defined similarly to the *outcome probability* variable, except that this variable refers to the future (Table 5.7).

Adjustment = {*Very low, Low, Medium, High, Very high*}

Let us presume that agent i has the goal X with "*very high*" *relevance*: $\mathbf{Rel}(\mathbf{Des}_i(X))$ = "very high". The *adjustment* variable measures the probability of X to occur at time t_{n+1}.

Adjustment	**Definition**
Very low	$0 \leq \mathbf{P}(X) \leq 0.2$
Low	$0.2 < \mathbf{P}(X) \leq 0.4$
Medium	$0.4 < \mathbf{P}(X) \leq 0.6$
High	$0.6 < \mathbf{P}(X) \leq 0.8$
Very high	$0.8 < \mathbf{P}(X) \leq 1$

Table 5.7: The values of the *adjustment* appraisal variable

Chapter 6

Implementation

This chapter presents all the information about implementing our *appraisal-based* model, including the game environment, BDI agent and the computational model as well. Figure 6.1 illustrates our implementation step-by-step. Below, we describe each of these steps in more detail.

1. The game environment (Battleship) is linked to a BDI agent. The agent constantly communicates with the game environment and exchanges information about the game state. This communication takes place in two ways. The agent perceives every change that occurs in the game, and updates its beliefs accordingly. Furthermore, the agent can affect the environment by taking actions.
2. The values of the appraisal variables will be derived from the agent's view of the game environment in terms of its beliefs, desires and intentions.
3. Scherer specifies the combination of the values of the appraisal variables in a mapping table. All these specified values are saved in an XML file and read out during the emotion elicitation process.
4. The computational model is used to determine which single emotion is triggered. The values of the appraisal variables are derived from the previous two phases. In order to illustrate how the computation works and for the sake of easier debugging, we have developed an interface.
5. The triggered emotion is visualized on the face of a virtual character.

6.1 Interaction between Battleship and BDI agent

The agent should know the same information that we would expect from a human player to know. The agent is coded in 2APL (pronounced "double a-p-l") agent programming language. The rest of the game is developed in JAVA (Figure 6.2). In this section, we will focus mainly on the implementation of the BDI agent.

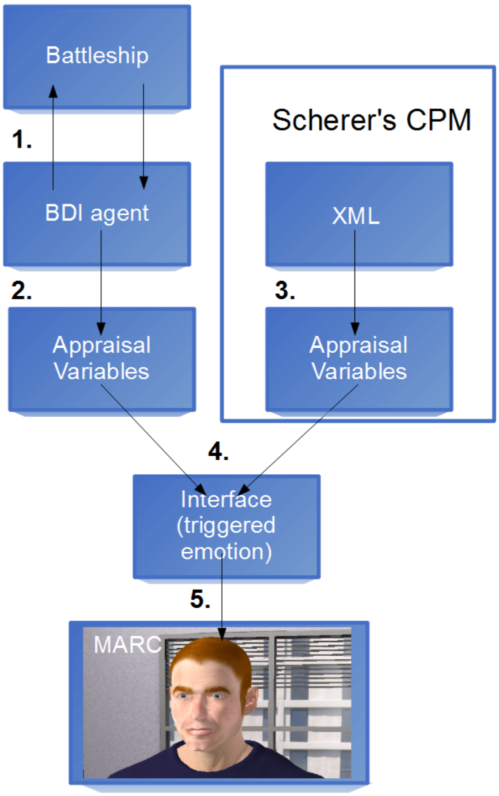


Figure 6.1: The overview of our implementation

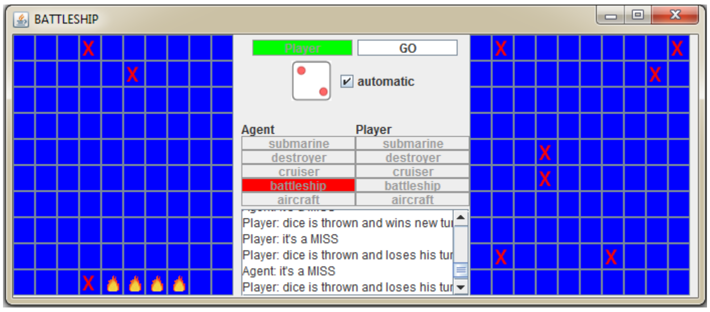
6.1.1 2APL agent in Battleship

There are many other BDI agent programming languages such as JACK or Jadex. However, we favour 2APL for our implementation because it was developed at Utrecht University as part of research projects [10].

The 2APL platform provides a graphical interface through, which users can load, run, and monitor the execution of 2APL multi-agent programs (Figure 6.3). It also integrates both declarative and imperative programming style. Goals and beliefs are written in declarative manner, whereas plans and external environments are implemented imperatively. These external environments are modular extensions that agents can have access to via external actions. External environments serve as an interface for JAVA programming language, which allows programmers to develop their own environment.

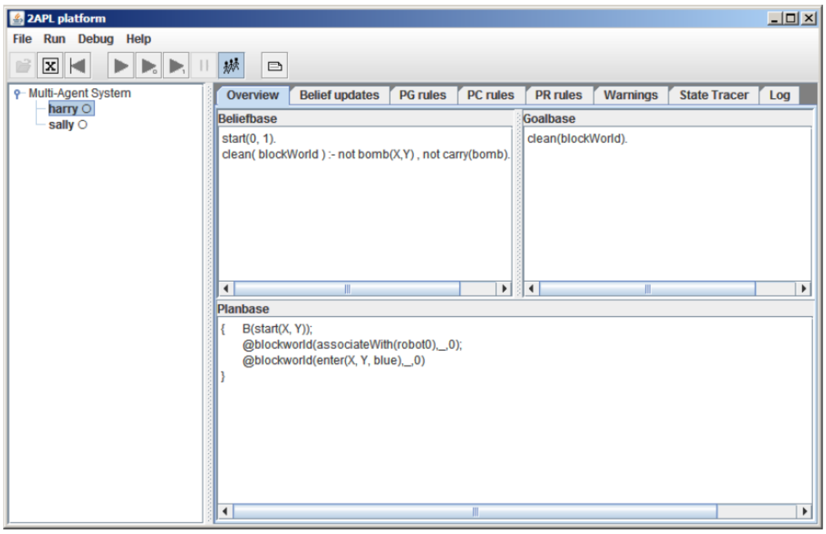
After introducing the agent programming language, we can break down the functionality of the Battleship game, so we will be able to specify the agent in 2APL. First, we need to define the possible actions that the agent may take:

π = {die throw, no die throw, select square}



```
Player: dice is thrown and wins new tu
Player: it's a MISS
Player: dice is thrown and loses his tu
Agent: it's a MISS
Player: dice is thrown and loses his tu
```

Figure 6.2: A screenshot of the JAVA game environment during the evaluation



```
start(0, 1).
clean( blockWorld ) :- not bomb(X,Y) , not carry(bomb).
```

```
clean(blockWorld).
```

```
{    B(start(X, Y));
     @blockworld(associateWith(robot0),_,0);
     @blockworld(enter(X, Y, blue),_,0)
}
```

Figure 6.3: The main user interface when selecting an agent (Harry) on the left (from [10])

Here is the list that contains all the events that may occur throughout a turn (emotion elicitation phase).

E = {miss a ship, hit a ship, sink a ship, lose turn, win turn *and* miss a ship, win turn *and* hit a ship, win turn *and* sink a ship, sink a ship *and* win game, win turn *and* sink a ship *and* win game }

Table 6.1 illustrates which type of consequences can actions trigger in Battleship.

Beliefs

Beliefs define the initial belief base of an agent. At runtime a belief base is used to represent the current state of the game environment. An emotion is triggered in the agent's turn and also in its opponent's turn because of the turn-based system of the game. This means that the agent must have beliefs about the actions and events of its opponent as well. Therefore, there will be different beliefs and goals of the agent in its own and in its opponent's turn. For

no die throw ∧ select square → miss a ship
no die throw ∧ select square → hit a ship
no die throw ∧select square → hit a ship ∧ sink a ship
die throw ∧ select square → win a turn ∧ miss a ship
die throw ∧ select square → win a turn ∧ hit a ship
die throw ∧ select square → win a turn ∧ hit a ship ∧ sink a ship
die throw → lose a turn
no die throw ∧ select square → hit a ship ∧ sink a ship ∧ win the game
die throw ∧ select square → win a turn ∧ hit a ship ∧ sink a ship ∧ win the game

Table 6.1: Actions and their consequences

example, in its own turn, it will desire to hit a ship, whereas in its opponent's turn, it will desire to avoid to be hit. In order to implement the game, the following information need to be specified as beliefs:

- whose turn it is
- whether the shot results in a miss, a hit or a sunk ship
- number of misses
- number of shots
- number of available cells
- number of sunk ships
- number of die tosses
- number of new turns gained by tossing the die

This information can be converted into 2APL beliefs (Table 6.2) and goals (Table 6.3). These beliefs and goals are updated everytime when something occurs in the game environment. However, these two tables represent only the agent's turn, which means that the same beliefs and goals need to be also specified for the agent in its opponent's turn.

Beliefs	**Description**
turn(*agent*)	The agent starts the game
numberOfTurns(*0*)	Number of turns played. Initially, it is 0.
numberOfMisses(*0*)	Number of missed shots. Initially, it is 0.
numberOfHits(*0*)	Number of hits. Initially, it is 0.
numberOfAvailableCells(*100*)	Number of available shots. Initially, it is 100.
numberOfSunkShips(*0*)	Number of sunk ships. Initially, it is 0.
numberOfDieTosses(*0*)	Number of die tosses. Initially , it is 0.
numberOfHighDieTosses(*0*)	Number of gained turns. Initially, it is 0.
win(*game*):- numberOfSunkShips(*5*)	Game is won if there is 5 sunk ships.

Table 6.2: Beliefs

Goals	**Description**
hit (*ship*)	The agent has the goal to hit the player's ships.
sink (*ship*)	The agent has the goal to sink a ship after hitting it.
throw (*highDice*)	The agent has the goal to throw the dice if it has fewer ships left than its opponent.
win (*game*)	The agent has the goal to win the game.

Table 6.3: Goals

6.2 Appraisal variables derived from BDI agent

In this section, we provide a possible interpretation of our previously introduced model in Battleship. It is likely that the previously defined generic meanings of the appraisal variables need to be modified to fit within the game.

6.2.1 Unexpectedness, outcome probability and discrepancy from expectation

This section covers three appraisal variables because all of them can be linked to probability but from different point of views. The *unexpectedness* variable measures whether the probability is greater or smaller than a subjective threshold, which is specified manually. The *outcome probability* variable represents the probability of the occurred event. Finally, the *discrepancy from expectation* variable measures the distance between the threshold and the probability of an event to occur.

The probability of the game is given by two components: the probability of a miss or a hit attempt (Figure 6.4), and the probability of tossing a high number with the die to gain a new turn (Figure 6.5).

```
$numberOfAvailableSquares = numberOfAvailableSquares - (numberOfHits + numberOfMisses)$
  if $shot = miss$ then
    $probabilityGrid = (83 - numberOfMisses)/numberOfAvailableSquares$
  else
    $probabilityGrid = (17 - numberOfHits)/numberOfAvailableSquares$
  end if
```

Figure 6.4: The probability value of missing, or hitting a ship

The numbers *83* and *17* represent the number of possible misses and hits accordingly. The number of available squares is *100* initially but after every attempt, it will be decreased. If there was a hit but the ship is not sunk yet then the probability will be set to *0.8.* This is a static value because it is not calculated in every turn, which implies that after hitting a ship, the likelihood that the agent hits the ship again is "*very high*".

If a player tosses the die, then the probability of the die needs to be added to our calculation as well. This additional information will slightly change our algorithm.

```
if $dice = tossed$ then
    if $dice = high$ then
        $probability = ((1 - (NumberOfDiceHigh/NumberOfTosses)) + probabilityGrid)/2$
    else
        $probability = NumberOfDiceHigh/NumberOfTosses$
    end if
else
    $probability = probabilityGrid$
end if
```

Figure 6.5: The previous algorithm is extended by the probability value of tossing a high number with the die

After calculating all the values for the variables, we need to assign a categorical label to them similarly, as we presented it in the generic appraisal model previously.

6.2.2 Goal Relevance

Our game scenario is special since there is always something goal relevant in every turn. The *goal relevance* variable is defined by the value of the most significant goal. However, in our game, we use the number of goals to determine the value of *goal relevance*. The goal of winning is achieved or failed then the *goal relevance* obtains "*very high*" label. Otherwise, the goals and their assigned values are defined in Table 6.4.

Value	**Goals**
Low	throw (*highDice*) / hit(*ship*) / sink(*ship*)
Medium	throw (*highDice*) and hit(ship) / hit(*ship*) and sink(*ship*)
High	throw (*highDice*) and hit(*ship*) and sink(*ship*)

Table 6.4: The value assignment of *goal relevance* in Battleship

6.2.3 Causal attribution

The *causal attribution* is implemented differently as it has been proposed in the previous chapter. In Battleship, all events can be connected to luck since until a ship is not hit, the agents shoot randomly. However, the agent can be also viewed as the cause of the events, when it tosses the die or attempts to sink a ship after it has already hit one. Thus, we link *causal attribution* to the goals of the game (Table 6.5).

Causal attribution	**Goals**
Chance	hit(*ship*)
Self / Other	throw(*highDice*) / sink(*ship* / win(*game*)

Table 6.5: The value assignment of *causal attribution* in Battleship

6.2.4 Goal conduciveness

The *goal conduciveness* determines the valence of an event in the Battleship game. For example, an event is *goal conducive*(takes a "*high*" value), if:

$$\mathbf{Bel}_i\ (\neg\ \text{hit}(\mathit{ship})) \wedge \mathbf{Des}_i\ (\text{hit}(\mathit{ship}))$$

6.2.5 Control and power

As opposed to the generic model, we define *control* variable slightly differently. Since our game is turn-based and *control* variable refers to the future, events will become controllable if the agent believes that the next turn will be its. The *power* variable also takes the same value as *control.*

Control	**Definition**
Low	at time t_{n+1}: $\mathbf{Bel}_i$ **(**turn(agent))
High	at time t_{n+1}: $\mathbf{Bel}_i$ **(**$\neg$ turn(agent))

Table 6.6: The value assignment of *control and power* variable in Battleship

6.2.6 Adjustment

The *adjustment* is defined differently than in the generic model, in which it measures whether the main goal can be achieved. In Battleship, to win the game, the agent has to sink 5 ships before its opponent. The value of the *adjustment* variable is determined by checking the difference between the number of remaining ships for both sides. If the agent believes that it has 3 ships left, while its opponent has 5, then the difference 2 will provide a categorical value such as "low" for the *adjustment* variable (Table 6.7).

Adjustment	**Difference**
Very low	1
Low	2
Medium	3
High	4
Very high	5

Table 6.7: The value assignment of *adjustment* variable in Battleship

6.3 Scherer's mapping table

In order to trigger emotions, we need to apply our previously introduced algorithm ("*Mapping from appraisal variables to emotions*") to measure similarity. This requires us to provide the precise combination of the values of the appraisal variables for each of the emotions that are implemented. Thus, we specify Scherer's mapping table in an XML file (Figure 6.6), which makes it easy to add, remove or modify emotions, and appraisal variables for future work.

```
<emotion>
    <type>happiness</type>
    <UnExpectedness>Open</UnExpectedness>
    <GoalRelevance>Medium</GoalRelevance>
    <CausalAttribution>Chance</CausalAttribution>
    <DiscrepancyFromExpectation>Low</DiscrepancyFromExpectation>
    <OutcomeProbability>Very High</OutcomeProbability>
    <GoalConduciveness>High</GoalConduciveness>
    <Control>Open</Control>
    <Power>Open</Power>
    <Adjustment>Open</Adjustment>
</emotion>
```

Figure 6.6: This is an example of specifying an emotion (happiness) in XML code

6.4 Emotion computation

In our research, the collection of the appraisal variables and their values are all stored in a so-called appraisal frame, which represents the current situation in the environment. In the Battleship game, an appraisal frame is equal a turn.

We have created a user interface that shows all the appraisal variables and their specified values on the scrollbars (Figure 6.7). The scales of the scrollbars are all categorical, displaying various values of the given appraisal variable. If one of the appraisal variables needs to be excluded from the computation, then it can be easily done by deselecting the radio button next to the scrollbar.

After every appraisal frame, the values of the appraisal variables are derived from the environment, and adjusted on the scrollbars. Based on the values, an emotion will be triggered. Furthermore, the users can also adjust the values by themselves and press on the compute button to trigger an emotion. This feature could be particularly useful for certain cases, when users want to deeply understand the way the computation works.

Our emotion triggering algorithm has three levels. All these levels are separated, and illustrated by various colors (Figure 6.8) for the user to better understand, when certain emotions get excluded during the computation (Figure 6.9).

The interface also provides scrollbars for users to personalize the computation by adjusting values for the threshold scrollbar. Depending on the value of the threshold variable, the *unexpectedness* and the *discrepancy from expectation*

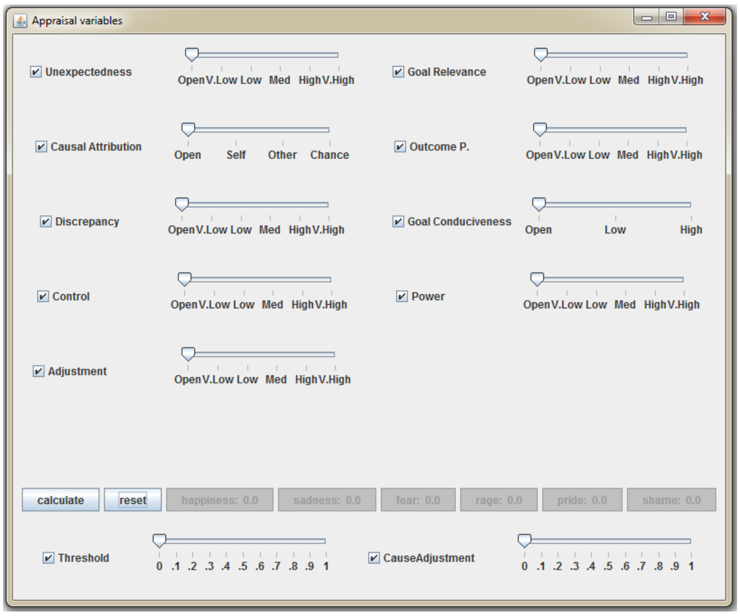


Figure 6.7: Interface of our emotion calculation

variables may take different values. This feature provides the feeling of giving a more subjective interpretation of situations to the agents.

In addition, users can also refine the search for the *causal attribution* variable. This scrollbar specifies the range, which helps to determine whether an emotion can remain candidate to be triggered or gets excluded from the computation.

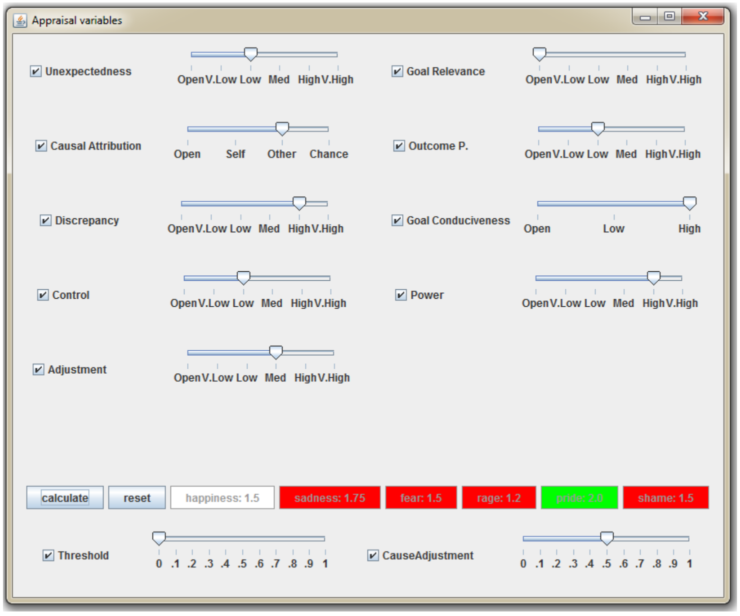
⬜ **triggered emotion**
🟥 **valence mismatch**
🟨 **valence match**
🟩 **causal refinement**
▨ **default**

Figure 6.8: The various colours represent different levels of emotion computation

6.5 Visualizing emotions

MARC is an MPEG-4 based facial animation system, which is capable of displaying body and facial animation in real-time. The application was developed by Matthieu Courgeon at LIMSI-CNRS universty, with the purpose of performing various user studies (Figure 6.10).

The application is easy to use, all we need to do is establishing the communication between our *appraisal-based* model and MARC. This communication



happiness: 1.5

sadness: 1.75

fear: 1.5

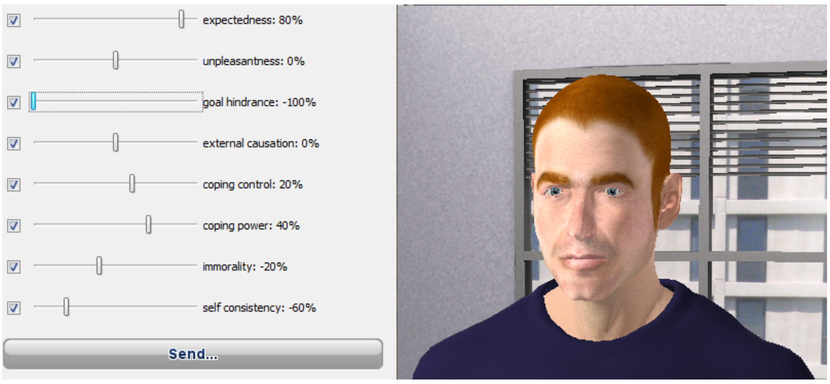
rage: 1.2

pride: 2.0

shame: 1.5

Figure 6.9: The user interface is shown during the computation. Based on the colour representation, sadness will be triggered.

takes place via UDP messages (an example of this message is provided in Appendix B). Courgeon et al. adapt different type of appraisal variables from Scherer's theory than we do in our research. However, we use this system only for making a demo to illustrate the benefits of our model. This means that there is no need for further investigation on how to find precise mapping between the two systems. We use only those appraisal variables that can be found in both works.



☑ expectedness: 80%

☑ unpleasantness: 0%

☑ goal hindrance: -100%

☑ external causation: 0%

☑ coping control: 20%

☑ coping power: 40%

☑ immorality: -20%

☑ self consistency: -60%

Figure 6.10: MARC platform

Chapter 7

Evaluation

Given the nature of our model, it is possible to generate quantitative and qualitative results as well. However, without the use of human data, the results can be only used to support claims about the system itself. Therefore, we have conducted several short informal interviews (See Appendix A) and asked what type of emotions the subjects feel if certain events occur in the game environment. Based on their answers, and our intuition, we have mapped the possible events of the Battleship game to emotions. The result of this mapping will provide the ground-truth for our evaluation, and henceforth, will be associated with *black-box* models. The following two tables below, contain the emotions and their triggering events. All the emotions are presented from the agent's aspect. Table 7.1 illustrates events and their emotions in the agent's turn, whereas Table 7.2 shows them in the opponent's turn.

The generated emotions and their eliciting events are saved after every turn during the game. We will use this data to evaluate whether our model meets our two expectations.

Firstly, inappropriate emotions should not be triggered in the game. This is difficult to measure since there are real-life situations when multiple type of emotions could be elicited in a person. Thus, as long as the valence of the given emotion is correct, we consider it appropriate. In other words, if an event occurs and our agent did not succeed in achieving its goal then a negative emotion should be triggered. Luckily, this feature has been already implemented in our computational model. Thus, such a fatal mistake will surely never occur.

Secondly, we expect our *appraisal-based* model to produce a richer set of emotions due to its dynamic nature. To give a short example, in Battleship, if the agent loses a ship, then it is configured to experience rage by assigning rage emotion to the losing ship event (*black-box* model). However, there might be more emotions with negative valence that would be appropriate for this particular situation, such as shame, fear or anger. In order to investigate this, we will compare the number and the type of emotions produced by both models.

7.1 Set-up

We thought of two possible set-ups for evaluation: *player versus agent* and *agent versus agent.* The first one is suitable for qualitative results because

Events(agent's turn from agent's aspect)	**Emotion**
Agent misses	Shame
Agent hits	Pride
Agent hits/sinks	Pride
Agent wins a turn and misses	Rage
Agent wins a turn and hits	Pride
Agent wins a turn and hits/sinks	Pride
Agent loses turn	Fear
Agent hits/sinks and wins game	Happiness
Agent wins a turn, hits/sinks and wins game	Pride

Table 7.1: The events and the assigned emotions from the agent's aspect in the agent's turn

Events(opponent's turn from agent's aspect)	**Emotion**
Opponent misses	Happiness
Opponent hits	Rage
Opponent hits/sinks	Rage
Opponent wins a turn and misses	Happiness
Opponent wins a turn and hits	Fear
Opponent wins a turn and hits/sinks	Fear
Opponent loses turn	Happiness
Opponent hits/sinks and wins game	Sadness
Opponent wins a turn, hits/sinks and wins game	Sadness

Table 7.2: The events and the assigned emotions from the agent's aspect in the opponent's turn

Description	**Evaluation**
Number of plays	14 (7-7) wins)
Number of events (agent-opponent)	958-920 turns
Threshold	0.5
Cause refinement	0.2

Table 7.3: The facts of the evaluation

players could influence the gameplay and, thus, the emotion generation. The second one is more suitable for quantitative results because the game is played automatically, so human players are not needed.

Conducting a further user study would not add much to our research, because we have already obtained the ground truth. Thus, we have made the decision to evaluate our work by letting the two agents play the game against each other. This implies that the game state cannot be influenced by the player. On the other hand, it can be run many times in a short period, which will produce a lot of data for analysis.

The set-up is fixed for all the runs, which means that the location of the agent's and its opponent's ships is the same every time, when the game is played. The agents are endowed with basic strategies. For example, if there is a hit, then the agents begin to look for the rest of the ship horizontally, and if the ship is not located that way, then vertically. Furthermore, they start tossing the die if they have fewer ships remained than their opponent, regardless the probability of their successful throws. This strategy implies that the event of "*agent/opponent wins a turn, hits/sinks and wins game*" can be excluded from our results. It is not possible to win the game with die tossing, since these two actions are mutually exclusive according to the strategy that our agents currently follow.

7.2 Results

There are two emotions with positive and four with negative valence. Therefore, we expect that the difference in the number of occurrences of emotions will be higher in the case of negative events.

Table 7.3 illustrates all the details about our evaluation. We have made sure that our evaluation will include the same amount of wins from both sides. Each side has had slightly less than 1000 emotion triggering events. In our opinion, this amount of runs should provide us enough information to be able to see whether our expectations are achieved. The two changeable variables on the interface were set to fixed values and remained that way throughout all the runs.

Table 7.4 presents the results obtained from the agent's turn. It is apparent from this table that all the events have triggered at least more than one emotion. In the case of negative events, certain emotions have dominated (Figure 7.1). For example, rage has been triggered approximately 10 times more than any other emotion when the "*agent misses*". The domination rate is almost the same when "*the agent tosses the die and misses*" but, in this case, shame was the dominating emotion. Interestingly, the situation is different when we investigate

Events(agent's turn)	***Black-box***	**Appraisal**
Agent misses	Shame: 380	Shame: 34 / Rage: 346
Agent hits	Pride: 92	Pride: 23 / Happiness: 69
Agent hits/sinks	Pride: 29	Pride: 22 / Happiness: 7
Agent wins a turn and misses	Rage: 136	Shame: 117 / Fear: 8 / Rage: 11
Agent wins a turn and hits	Pride: 60	Pride: 29 / Happiness: 31
Agent wins a turn and hits/sinks	Pride: 25	Pride: 17 / Happiness: 8
Agent loses turn	Fear: 229	Shame: 155 / Sadness: 28 / Fear: 46
Agent hits/sinks and wins game	Happiness: 7	Pride: 3 / Happiness: 4
	958	

Table 7.4: The events, their assigned and appraised emotions with their occurrences from the agent's aspect in the agent's turn

the "*agent loses a turn*" event. The number of occurrences is more even than in the previous two examples. It can be seen from the data in Table 7.4 that positive emotions are triggered more or less evenly, except when *"agent hits*" and when "*the agent sinks*", but the dominating rate never goes beyond 3.

Figure 7.2 indicates the number of occurrences per emotion type. What is interesting in this data is that all the emotions have been triggered. It is also interesting to see the degree of domination of shame and rage, compared to the other two negative emotions. This figure also reveals that the likelihood of triggering negative emotions is much higher than positive emotions. The reason is that we have a strict definition of *goal conduciveness* variable. The chance of achieving goals is much lower than failing them in the agent's turn. This should be precisely the opposite in the data obtained from the opponent's turn.

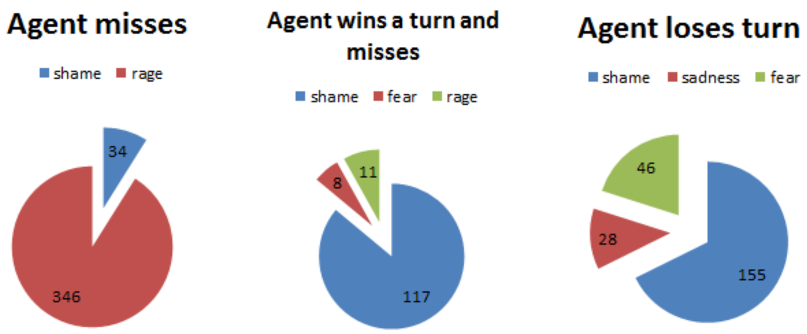


Figure 7.1: The number of occurrences of various emotion types in the case of different events. 1. "*agent misses*" (left) 2. "*the agent tosses the die and misses*" (middle) 3. "*agent loses a turn*" (right)

Table 7.5 shows the results obtained from the opponent's turn. In these turns, the positive emotions dominate as opposed to the agent's turn. Happiness is triggered way more than any of the other emotions. For example, the "*opponent loses turn*" triggers happiness 80 times more than pride. As a matter of fact, in the case of this particular event, this behaviour is appropriate, since the opponent's turn losing should not be caused by the agent. Similarly to hap-

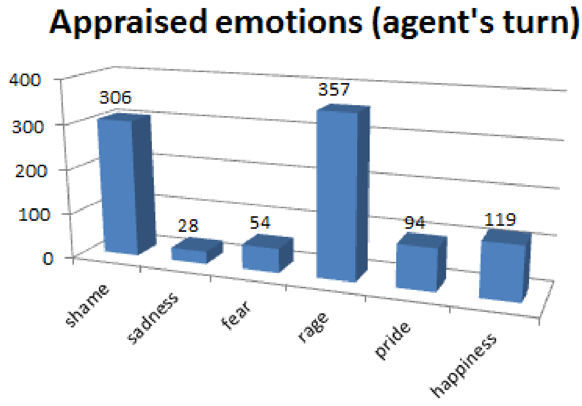


Figure 7.2: The number and the type of emotions in the agent's turn generated by the appraisal-based model

Events(opponent's turn)	**Black-box**	**Appraisal**
Opponent misses	Happiness: 429	Pride: 148 / Happiness: 281
Opponent hits	Rage: 94	Shame: 12 / Fear: 82
Opponent hits/sinks	Rage: 27	Shame: 5 / Fear: 22
Opponent wins a turn and misses	Happiness: 108	Pride: 20 / Happiness: 88
Opponent wins a turn and hits	Fear: 56	Shame: 4 /Sadness: 5 / Fear: 47
Opponent wins a turn and hits/sinks	Fear: 26	Shame: 4 / Fear: 22
Opponent loses turn	Happiness: 173	Pride: 2 / Happiness: 171
Opponent hits/sinks and wins game	Sadness: 7	Sadness: 1 / Fear: 6
	920	

Table 7.5: Events, their assigned and appraised emotions with their occurrences from agent's aspect in opponent's turn

piness, fear dominates among the negative emotions. However, there are cases when, in our opinion, fear is not the most appropriate emotion. For example, when the *"opponent hits/sinks and wins the game"* then sadness would be a better choice.

Figure 7.3 provides an overview about the occurrences of all the emotions in the opponent's turn. This figure reveals that rage has not been generated at all during the opponent's turns. Furthermore, happiness has been produced 2.5 times more than pride and more than half of the events has elicited happiness. The dominating rate is even higher among negative emotions than between pride and happiness.

Comparing the number of emotions of both turns triggered by our model to the *black-box* model (Figure 7.4) highlights that a richer set of emotions has been elicited by our model. However, the triggered emotions would need further analysis to understand why certain emotions are elicited significantly more than others. This evaluation could be part of future work.

The current game environment has a limited number of actions that a char-

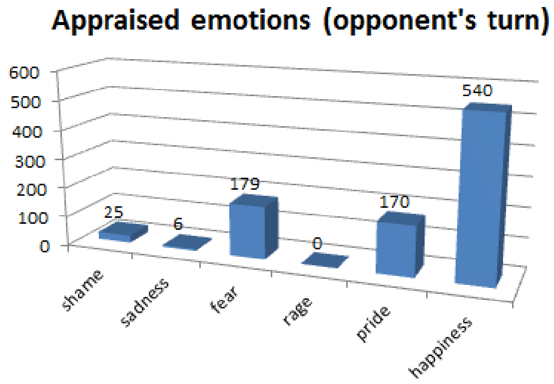


Figure 7.3: The number and type of the generated emotions in the agent's turn by the appraisal-based model

acter can take, which limits the number of events. We have preferred to choose such a game because, this way, keeping track of the appraisal process is much easier. However, we would expect a complex game, which has more unpredictable events than our simple game to produce more significant results.

In our opinion, the main difference between the two models lies in appraisal variables because they provide the subtle differences between similar emotions. Applying them enables game creators to trigger emotions in NPCs with similar manner as emotions are triggered in humans.

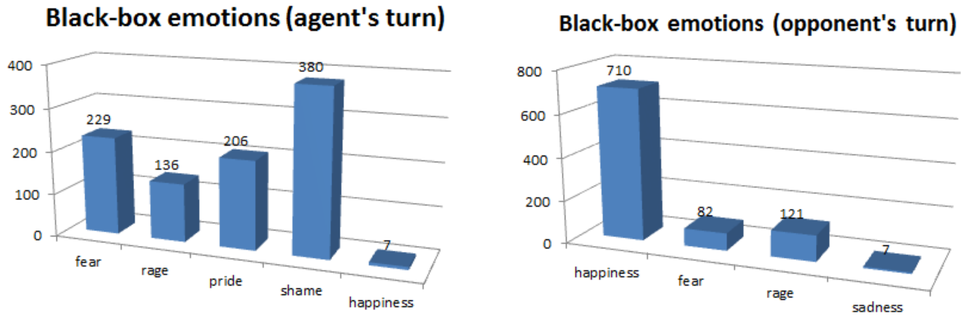


Figure 7.4: The number and type of the generated emotions in the agent's and in the opponent's turn by the *black-box* model

Chapter 8

Conclusion

We have accomplished to develop an *appraisal-based* model that is capable of emotion generation in NPCs without the need of assigning emotions to events manually. This emotion elicitation process takes place similarly as it does in humans since our work is based on a well-recognised psychological research.

Our research can be broken down into smaller independent components. Firstly, we had found a simple game based on several criteria, which was used to evaluate our model. Secondly, we have proposed a computational model that is suitable for sorting emotions according to their valence, and trigger emotions by comparing their similarity to Scherer's emotions. Thirdly, we have provided semi-formal definitions for the appraisal variables, so they can obtain values from a game environment.

Our current agent implementation is based on an ad-hoc solution. However, as a future work, an agent programming language could be extended to automatically trigger emotions, as it has been already done for the OCC model before [33]. Throughout our research, we have attempted to remain as generic as possible, which implies that any of these three components would probably work independently from the rest. One great disadvantage of our model is that it always requires probability for every change that occurs in the game environment. This can be quite difficult if these changes are not predictable.

Our evaluation has shown that our model is capable of producing at least as good results as *black-box* models. With our model, a broader range of emotion types could be triggered in an NPC, which subsequently would provide more credibility. A character with the capability of expressing various positive emotions - if a certain positive event occurs - would be definitely more believable than a character, which always displays happiness.

In conclusion, our *appraisal-based* model can provide a great help for game creators that long for endowing their NPCs with dynamically generated emotions. However, there are still many ways to improve our model, which we will discuss in the next chapter.

Chapter 9

Future work

In this chapter, we discuss possible ways of improving our research and making it more complete. At the moment, a few of our appraisal variables take only a limited number of values, which does not correspond to Scherer's theory. *Unexpectedness, goal conduciveness* and *control* can be only "low" or "high", while in the original specification they can take all type of values. Furthermore, the rest of the appraisal variables could be implemented as well. Other factors such as preference (*pleasantness*), time (*urgency*), social norms and values (*internal and external standards*), etc. would be also taken into account. Implementing all the appraisal variables and the possibility to assign all the five values would probably result in a greater diversity in the number of occurrences of the emotions.

For the sake of simplicity, triggering multiple emotions - as a response for an event - is not possible in our work. However, in real life, several emotions may coexist simultaneously. For example, a person can be happy about winning a game but feeling ashamed the way he did.

We have discussed smaller improvements above. There are many other features that could be implemented. Our computational model could be modified, so the computation takes place differently. Instead of discrete time, continuous time could be used, which would require us to define more precisely when an emotion needs to be triggered. All these features are related to the appraisal stage. In the case of a more complex model, the remaining two stages (emotion intensity and effects) should be added as well.

Intensity could be measured by the degree of similarity of emotions. This information is already available in our computational model but not yet used. In our current model, if an emotion is triggered, then the values of the appraisal variables will not be added to the next computation, which means that emotions are triggered from scratch. A more realistic way would require a smooth transition between emotional states. This could be implemented by a new affective state, namely mood: a sad person will react to a positive event fairly differently than a happy person. Furthermore, ramp-up and decay of the emotion intensity over time could be added to our model as well. Unfortunately, Scherer does not share any details on these subjects.

Implementing the emotion effects would require two distinct classes of processes. The visible effects are facial expressions, gestures, speech and movements. The internal effects affect the cognitive processes, which will subsequently influence the agent's decision making and behaviour as well. An extensive base of data is available to develop the visible effects. However, the situation is quite different when it comes to defining the mapping between an emotion and its effects on cognitive processes.

Bibliography

[1] C. Adam, B. Gaudou, A. Herzig, and D. Longin. Occ's emotions: A formalization in a bdi logic. In J. Euzenat and J. Domingue, editors, *Artificial Intelligence: Methodology, Systems, and Applications*, volume 4183 of *Lecture Notes in Computer Science*, pages 24–32. Springer Berlin Heidelberg, 2006.

[2] M. B. Arnold. *Emotion and personality (Vol 1). Psychological aspects.* New York: Columbia University Press, 1960.

[3] C. Becker-Asano. *WASABI: Affect Simulation for Agents with Believable Interactivity.* PhD thesis, Faculty of Technology, University of Bielefeld, 2008. IOS Press (DISKI 319).

[4] C. Becker-Asano and I. Wachsmuth. Affective computing with primary and secondary emotions in a virtual human. *Autonomous Agents and Multi-Agent Systems*, 20:32–49, 2010.

[5] M. E. Bratman. Intentions, plans, and practical reason. Harvard University Press, 1987.

[6] J. Broekens, D. DeGroot, and W. A. Kosters. Formal models of appraisal: Theory, specification, and computational model. *Cognitive Systems Research*, 9(3):173 – 197, 2008.

[7] M. Courgeon, C. Clavel, and J.-C. Martin. Appraising emotional events during a real-time interactive game. In *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*, AFFINE '09, pages 7:1–7:5, New York, NY, USA, 2009. ACM.

[8] A. R. Damasio. *Descartes' error, emotion reason and the human brain.* New York: Avon Books, 1994.

[9] C. Darwin. *The expression of the emotions in man and animals.* New York: D. Appleton and company. (received electronically from Electronic Text Center, University of Virginia Library, first published in 1872), 1898.

[10] M. Dastani. 2apl: a practical agent programming language. *Autonomous Agents and Multi-Agent Systems*, 16(3):214–248, June 2008.

[11] A. Egges. *Real-time Animation of Interactive Virtual Humans.* PhD thesis, Faculte des Sciences, University of De Geneve, 2006.

[12] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, California, 1978.

[13] J. Gratch and S. Marsella. A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, 5:269–306, 2004.

[14] E. Hudlicka. Affective computing and game design. In *In Proceedings of the 4th Intl. North American Conference on Intelligent Games and Simulation*, pages 5 – 12, Montreal, Canada, 2008. McGill University.

[15] W. James. *What is an emotion?* Mind, 9, 188-205, 1884.

[16] C. Lange. *Om Sindsbevoegelser: Et psykofysiologiske Studie.* Kopenhagen: Kronar.(deutsch 1887: Ueber Gemuthsbewegungen. Leipzig: Theodor Thomas).

[17] R. S. Lazarus. *Emotion and Adaptation.* New York: Oxford University Press, 1991.

[18] L. Malatesta, A. Raouzaiou, K. Karpouzis, and S. D. Kollias. Towards modeling embodied conversational agent character profiles using appraisal theory predictions in expression synthesis. *Applied Intelligence*, 30(1):58–64, Feb. 2009.

[19] R. P. Marinier and J. E. Laird. Computational modeling of mood and feeling from emotion. In *In Proceedings of the 29th Annual Conference of the Cognitive Science Society. Cognitive Science Society*, 2007.

[20] R. P. Marinier, J. E. Laird, and R. L. Lewis. A computational unification of cognitive behavior and emotion. *Cognitive Systems Research*, 10(1):48 – 69, 2009. Modeling the Cognitive Antecedents and Consequences of Emotion.

[21] A. Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14:261–292, 1996.

[22] J.-J. C. Meyer. Reasoning about emotional agents: Research articles. *Int. J. Intell. Syst.*, 21:601–619, June 2006.

[23] K. Oatley and J. M. Jenkins. *Understanding Emotions.* Blackwell Publishing, 1996.

[24] M. Ochs, K. Devooght, C. Pelachaud, and D. Sadek. A computational model of capability-based emotion elicitation for rational agent. 2006.

[25] M. Ochs, R. Niewiadomski, C. Pelachaud, and D. Sadek. Intelligent expressions of emotions. In *ACII*, pages 707–714, 2005.

[26] A. Ortony, A. Collins, and G. L. Clore. *The cognitive structure of emotions.* Cambridge: Cambridge University Press, 1988.

[27] M. Paleari, A. Grizard, and C. L. Lisetti. Adapting psychologically grounded facial emotional expressions to different anthropomorphic embodiment platforms. In *FLAIRS Conference*, pages 565–570, 2007.

[28] C. Pelachaud. *Modelling Multimodal Expression of Emotion in a Virtual Agent.* Philosophical Transactions of Royal Society B Biological Science, 2009.

[29] A. Rollings and E. Adams. *Andrew Rollings and Ernest Adams on Game Design.* New Riders Games, 2003.

[30] K. R. Scherer. On the nature and function of emotion: a component process approach. In K. R. Scherer and P. Ekman, editors, *Approaches to emotion*, pages 293–317. L. Erlbaum Associates, 1984.

[31] K. R. Scherer. *Appraisal considered as a process of multi-level sequential checking*, pages 92–120. Oxford University Press, New York and Oxford, k. r. scherer, a. schorr, and t. johnstone (eds.) edition, 2001.

[32] H. Schlosberg. A scale for judgement of facial expressions. *Journal of Experimental Psychology*, 29:497 − 510, 1954.

[33] B. R. Steunebrink. *The Logical Structure of Emotions.* PhD thesis, Utrecht University, The Netherlands, 2010.

[34] F. Thomas and O. Johnston. *Understanding Emotions.* The Illusion of Life: Disney Animation, 1981.

Appendix A

Questionnaire

1. Play the game first:

   a, Place you ships on the grid, you can rotate your ships with the right mouse click. There is also a reset button to start the placing all over and a finished which indicates that the placing is done.

   b, The basic Battleship rules and instructions for playing the game are each player calls out one shot (or coordinate) each turn in attempt to hit one of their opponent's ships. First, you must choose whether you would like to throw the die. If you do but throw 1,2,3 then you will lose the turn, so it is your opponent's turn again. However, if you throw 4,5,6 then you have earned a new turn. To "hit" one of your opponent's ships, you must click on the grid where you think one of their ships is located. The instructions state that once a shot is called, the opponent must immediately call out "hit" or "miss". If one of your ships gets hit, place a red peg over the hole location on your ships that was called out. If calling a shot (or trying to hit your opponent's ships), mark a red peg (if a hit was made) or a green peg (a miss) on your target grid located on the lid or the vertical divider between you and your opponent. This will help you keep track of your hits and misses in your hunt to find their ships.

   Once all holes on a ship have been filled with red pegs, your ship has sunk and must be removed from the ocean. You then announce which ship has sunk. The Battleship rules on successfully sinking a ship are as follows: Aircraft carrier - 5 hits, Battleship - 4 hits, Destroyer - 3 hits, Submarine - 2 hits.

2. Please choose ONE or MORE of the emotional states below which you have experienced while one of the following events occurred (If the event did not occur then IMAGINE how you would have felt about it). Please assign one of the following intensity labels to your selected emotion (Very low, Low, Medium, High, Very high).

   Emotions:

   Happiness, Joy, Displeasure, Contempt, Sadness, Despair, Anxiety, Fear, Irritation, Rage, Boredom, Shame, Guilt, Pride.

   - You miss a ship

- You hit a ship
- You hit/sink a ship
- You win a new turn and miss a ship
- You win a new turn and hit a ship
- You win a new turn and hit/sink a ship
- You lose a turn
- You hit/sink a ship and win game
- You win a new turn and hit/sink a ship and win game

- Your opponent misses a ship
- Your opponent hits a ship
- Your opponent hits/sinks a ship
- Your opponent wins a new turn and misses a ship
- Your opponent wins a new turn and hits a ship
- Your opponent wins a new turn and hits/sinks a ship
- Your opponent loses a turn
- Your opponent hits/sinks a ship and wins game
- Your opponent wins a new turn and hits/sinks a ship and wins game

Appendix B

MARC XML code

```
 message =
"<emotionml>\n"+
"  <emotion>\n"+
"    <appraisals set=\"scherer_cpm_checks\" >\n"+
"      <expectedness       value=\"0\"/>\n"+
"      <unpleasantness     value=\"1\"/>\n"+
"      <goal_hindrance     value=\"1\"/>\n"+
"      <external_causation value=\"1\"/>\n"+
"      <copying_control    value=\"0\"/>\n"+
"      <copying_power      value=\"0\"/>\n"+
"      <immorality         value=\"0\"/>\n"+
"      <self_consistency   value=\"0\"/>\n"+
"    </appraisals>\n"+
"  </emotion>\n"+
"</emotionml>\n";
```