# Chapter 13
# Emotional Appraisal Engines for Games

**Joost Broekens, Eva Hudlicka, and Rafael Bidarra**

**Abstract** Affective game engines could support game development by providing specialized emotion sensing (detection), emotion modeling, emotion expression and affective behavior generation explicitly tailored towards games. In this chapter we discuss the rationale for specialized *emotional appraisal engines* for games, analogous to having specialized physics engines. Such engines provide basic emotion modeling capabilities to generate emotions for Non Player Characters (NPCs), just like the *Havok* engine provides physics-related special purpose processing. In particular, such engines provide NPCs with an emotional state by simulating the emotional meaning of an event to an NPC in the context of the game's storyline, the NPC's personality, and relationships with other NPCs. We discuss why such engines are needed, present an example approach based on cognitive appraisal, and show how this appraisal engine has been integrated in a wide variety of architectures for controlling NPCs. We conclude with a discussion of novel gameplays possible by the more sophisticated emotion modeling enabled by an emotion appraisal engine.

## Introduction

Emotions are arguably the most important element of gaming and game design [1–6]. Players experience a wide range of emotions during the entire gameplay process. This begins even before playing the game in the form of anticipation (hope) raised by advertisement or previous experiences and disappointment or confirmation of the hope raised by developer blogs about the expected game features. Then the process of ordering (unpacking) and installing involves a wide variety of affective experiences including frustration, eagerness, anger, relief, and concentration. The first actual contact with the game can involve feelings of awe and belonging (e.g. in

J. Broekens (✉) • R. Bidarra
Intelligent Systems Department, Delft University of Technology, Delft, The Netherlands
e-mail: Joost.broekens@gmail.com; R.Bidarra@tudelft.nl

E. Hudlicka
College of Computer Science, University of Massachusetts-Amherst & Psychometrix Associates, Amherst, MA, USA
e-mail: hudlicka@cs.umass.edu

K. Karpouzis, G.N. Yannakakis (eds.), *Emotion in Games*,
Socio-Affective Computing 4, DOI 10.1007/978-3-319-41316-7_13

the case of a well-designed sequel). Thus by the time we are actually playing the game, we have gone through an important process of emotional investment in the game. This creates a bond between the game and the player, sometimes so strong that this bond can have serious negative consequences for the player and his/her surroundings. On the other hand that same bond creates a willingness to invest time and effort in playing the game and generates unique positive experiences not possible with any other media.

All four of the core areas of affective computing can contribute to a more immersive and engaging gameplay experience: emotion sensing and recognition, emotion modeling, emotion-driven Non Player Character (NPC) behavior and expression of emotions by NPCs, as well as affective player modeling. Other chapters in this volume discuss emotion sensing and recognition in gaming as well as emotion-driven expression and NPC behavior, and affective content. This chapter focuses on emotion modeling proper, that is, the simulation of emotions for the Non Player Characters in a game. “Simulating emotions” refers here to the process of automatically determining when an NPC should express (or behave according to) a particular emotion and with what intensity. This includes two broad categories of underlying processing: *generation of NPC emotion* in response to some set of emotion-eliciting stimuli within the gameplay, and *modeling of the effects of these emotions* on the NPCs’ internal processing (e.g., perception, decision-making, planning) and, ultimately, behavioral choices. In other words, we refer to computational modeling of emotion in the sense proposed by Hudlicka [7, 8]. In this chapter we focus on the first process: the generation of emotion in response to emotion-eliciting stimuli, called *appraisal*. Further, we focus on the “when to, what to, and how much to express”, not on the “how to express and what to do”; i.e., in this chapter we are not concerned with rendering emotional expressions or generating particular behavior that should follow particular emotions (e.g., fear followed by the behavior of fleeing). These aspects of emotional processing fall within emotion expression processing and are addressed in other chapters in this volume.

NPCs in many existing games do possess emotional behavior (expression, actions). However, these emotions are typically scripted, event-triggered, or built-in within the storyline. As such, emotions in NPCs are more often considered a cosmetic feature, related to rendering and behavioral realism, rather than an integral component of the game mechanics that can influence gameplay. The lack of emotional complexity and flexibility in existing NPCs severely limits their affective realism, and, consequently, the realism and immersive potential of the resulting gameplay. To address this limitation, NPCs need to include deeper models of affective processing, as outlined above. Such model-based emotions can simulate plausible emotional reactions at appropriate moments during gameplay automatically, thereby increasing the believability of the non-player character [9, 10] by increasing the variability of NPC behavior [11] and enabling novel forms of gameplay [12–14]. For a more in-depth analysis of the rationale for including deeper models of emotions in NPCs, as well as existing approaches, see [3, 15, 16]. By “plausible emotional reaction” we mean that the emotion of the NPC should make sense to the player; i.e., the emotion should be *psychologically valid*, within the context of the gameplay. Note that psychological validity does not imply that the emotion must be "normal", according to what normal individuals would feel in a particular setting. Rather, by a *psychologically valid model* we mean that *NPC emotions are simulated by a model that is based on what is known about human emotional appraisal*. This includes the possibility of generating unusual, abnormal or pathological characters, including plain evil characters, as these often fulfill important roles in a game's narrative.

In this chapter we discuss why specialized *emotional appraisal engines* for games are needed (section "Why Are Model-Based NPC Emotions Rare in Commercial Games?") and possible (sections "Emotional Appraisal Engines as Plug-in Modules", "Integrating Emotional Appraisal Engines with NPC Control", and "Appraisal Engines Enable Novel Gameplays and Genres"). The term 'appraisal engine' refers to specialized game engines that support modeling of emotions in NPCs, in a manner that does not require a commitment to a particular NPC architecture. In other words, emotional appraisal engines provide a modular approach to augmenting NPCs with emotion, analogous to a plug-in.

The term 'appraisal engine' contrasts with the term 'affective game engine' [11]. As originally envisioned, affective game engines would provide a broad range of tools to support the development of affective and affect-adaptive games by providing functionalities to facilitate implementing all four of the core areas of affective computing: recognition of player emotions, emotion modeling in NPCs (including both emotion generation and modeling of emotion effects on the NPCs internal processing and behavior), expression of emotions by NPCs, and affective player modeling. Currently, no game engine exists that provides all of these functionalities.

While many computational models of emotion are available [17–19] that could be used in games and experimental games exist that use some of these models [13, 20], commercial games have not incorporated model-based emotions (with some exceptions, see [3, 15]). We believe this is due to several reasons including: challenges associated with integrating cognitive appraisal models with existing NPC AI; lack of standard testing and development tools, resulting in perceived high risk and development costs; the need to understand the mechanisms mediating emotion elicitation; lack of examples of novel gameplays enabled by emotional NPCs; and, the conviction that gamers don't ask for emotional NPCs.

Support for these arguments can be found in a recent pilot study in which we investigated game designers' perspectives on emotional characters. Results from this study will be discussed in more detail in section "Why Are Model-Based NPC Emotions Rare in Commercial Games?". To show how these causes can be mitigated, we discuss recent projects in the area of development and integration of emotional appraisal engines. In section "Emotional Appraisal Engines as Plug-in Modules", we briefly present a recent approach towards developing an emotional appraisal engine [15]. We show how this approach addresses several of the issues outlined above, specifically: how it eliminates the dependency between existing NPC AI and the mechanisms required for dynamic emotion simulation; how it reduces the need to understand the emotion elicitation process; and how it enables control over the emotional behavior of NPCs. In section "Integrating Emotional

Appraisal Engines with NPC Control" we show that proper encapsulation of emotional appraisal enables integration with widely different NPC control mechanisms. In section "Appraisal Engines Enable Novel Gameplays and Genres", we present examples of novel gameplays enabled by NPCs augmented with model-based emotions. Some of these novel types of gameplays can be considered novel genres [14] that go beyond the traditional emotional deepening of the relationship between the player and the NPC [5].

## Why Are Model-Based NPC Emotions Rare in Commercial Games?

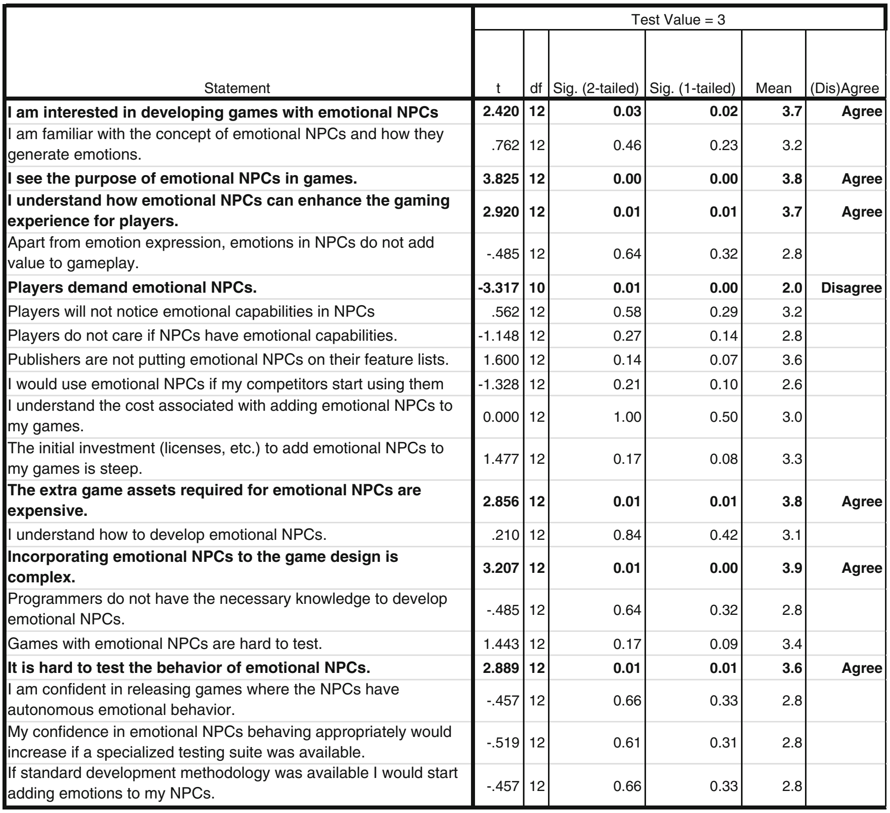
We believe there are several reasons why NPCs with emotions based on a computational model are rare in commercial games. These reasons are technical, conceptual and financial (see list below). These reasons can also been seen as a list of concerns regarding the feasibility of implementing model-based emotions in NPCs, and thus provide a set of requirements for affective game engines in general, and appraisal engines in particular.

1. Most emotion models are dependent on particular NPC AI: *complexity and modularization* (*technical*)
2. Lack of tools for design, development, and testing: *dev. support* (*technical*)
3. Complexity of modeling emotion elicitation: *complexity of emotions* (*conceptual*)
4. Lack of emotion-enabled gameplay innovation: *gameplay* (*conceptual*)
5. Players do not demand emotional NPCs: *market demand* (*financial*)
6. Waiting for a big game publisher to bite the bullet: *financial risk* (*financial*)

From a technical point of view, emotion modeling is rarely adapted to the game development process. Even though emotion modeling, especially based on the Ortony, Clore and Collins model (OCC) [21], has been used [22] and analyzed extensively [23–25] in research, none of these approaches have led to emotion simulation in games by means of computational modeling.

A major reason is that emotion modeling in research usually adopts a particular type of Artificial Intelligence NPC architecture, with its associated representational and reasoning mechanisms (e.g., fuzzy logic, BDI-agents), as basis for the computational model of emotion. However, just as a game developer does not want its destroyable assets to follow a particular morphology when they use a special purpose physics engine to simulate destruction, game developers also do not want to use a particular type of AI for their agents when simulating emotions. These aspects of game design need to be, and can be, decoupled [15]. *A successful appraisal engine thus needs to be an AI-independent module*, *with a clear Application Programming Interface* (*API*).

A second major reason is that developing a game involves a great deal of design and testing and the tools supporting these activities are lacking. It is also unclear how one should go about debugging emotions of NPCs in the first place. What kind of visualizations are needed? What kind of aggregation over NPCs and over time is needed to quickly get an idea of what happens emotionally to a set of NPCs? What is the exact behavior of a particular computational model in different settings? Just as a physics engine has predictable behavior with regards to physics (at least to a certain extent), an emotion simulation engine also needs some way of giving the game designer a sense of control over the resulting NPC behaviors. The difficulties associated with testing of emotional behavior of NPCs have also been confirmed with our pilot study (Fig. 13.1). *A successful appraisal engine needs to incorporate design and testing tools that give the game developer insight into the range of*



| | Test Value = 3 | | | | | |
|---|---|---|---|---|---|---|
| Statement | t | df | Sig. (2-tailed) | Sig. (1-tailed) | Mean | (Dis)Agree |
| **I am interested in developing games with emotional NPCs** | **2.420** | **12** | **0.03** | **0.02** | **3.7** | **Agree** |
| I am familiar with the concept of emotional NPCs and how they generate emotions. | .762 | 12 | 0.46 | 0.23 | 3.2 | |
| **I see the purpose of emotional NPCs in games.** | **3.825** | **12** | **0.00** | **0.00** | **3.8** | **Agree** |
| **I understand how emotional NPCs can enhance the gaming experience for players.** | **2.920** | **12** | **0.01** | **0.01** | **3.7** | **Agree** |
| Apart from emotion expression, emotions in NPCs do not add value to gameplay. | -.485 | 12 | 0.64 | 0.32 | 2.8 | |
| **Players demand emotional NPCs.** | **-3.317** | **10** | **0.01** | **0.00** | **2.0** | **Disagree** |
| Players will not notice emotional capabilities in NPCs | .562 | 12 | 0.58 | 0.29 | 3.2 | |
| Players do not care if NPCs have emotional capabilities. | -1.148 | 12 | 0.27 | 0.14 | 2.8 | |
| Publishers are not putting emotional NPCs on their feature lists. | 1.600 | 12 | 0.14 | 0.07 | 3.6 | |
| I would use emotional NPCs if my competitors start using them | -1.328 | 12 | 0.21 | 0.10 | 2.6 | |
| I understand the cost associated with adding emotional NPCs to my games. | 0.000 | 12 | 1.00 | 0.50 | 3.0 | |
| The initial investment (licenses, etc.) to add emotional NPCs to my games is steep. | 1.477 | 12 | 0.17 | 0.08 | 3.3 | |
| **The extra game assets required for emotional NPCs are expensive.** | **2.856** | **12** | **0.01** | **0.01** | **3.8** | **Agree** |
| I understand how to develop emotional NPCs. | .210 | 12 | 0.84 | 0.42 | 3.1 | |
| **Incorporating emotional NPCs to the game design is complex.** | **3.207** | **12** | **0.01** | **0.00** | **3.9** | **Agree** |
| Programmers do not have the necessary knowledge to develop emotional NPCs. | -.485 | 12 | 0.64 | 0.32 | 2.8 | |
| Games with emotional NPCs are hard to test. | 1.443 | 12 | 0.17 | 0.09 | 3.4 | |
| **It is hard to test the behavior of emotional NPCs.** | **2.889** | **12** | **0.01** | **0.01** | **3.6** | **Agree** |
| I am confident in releasing games where the NPCs have autonomous emotional behavior. | -.457 | 12 | 0.66 | 0.33 | 2.8 | |
| My confidence in emotional NPCs behaving appropriately would increase if a specialized testing suite was available. | -.519 | 12 | 0.61 | 0.31 | 2.8 | |
| If standard development methodology was available I would start adding emotions to my NPCs. | -.457 | 12 | 0.66 | 0.33 | 2.8 | |

**Fig. 13.1** (Dis-) agreement with statements about emotional NPCs based on 13 subjects (5 international, 8 Dutch) from different game development companies. Subjects rated agreement on a 5-point scale. (Dis-) agreement (*bold rows*) was decided based on statistical significance of a 1-tailed T-test (test value = 3)

*emotions*, *the intensity of emotions*, *and the causes for emotions at different levels of aggregation* (*individual NPCs*, *groups of NPCs*, *and even game world areas*).

For non-specialists, it is difficult to understand what emotion modeling really is, and what it can bring to the gameplay. For example, if one assumes that NPC emotions serve to evoke deep player-NPC interaction (an opinion explicitly voiced in our pilot experiment by one of the participants, as well as in [5]), then of course modeled NPC emotions are limited to adventure and RPG-like genres. However, this is not the case. Even simple puzzle games can benefit from, or even completely revolve around, emotion simulation ([14], and section "Appraisal Engines Enable Novel Gameplays and Genres"). If one assumes that modeled emotions are always psychologically plausible, and that this means that the emotions are those of healthy and sane individuals, this excludes emotion modeling for antagonists, as these characters usually are mean, deranged, mad, or otherwise not normal. It also excludes the possibility of developing serious games for psychotherapy, which may require the modeling of abnormal affective reactions and behavior motivated by emotion dysregulation. However, this is also not the case. Computational models of emotion are based on an understanding of affective processes based on psychological emotion theories, and these theories can also be applied to simulating "bad guys" in entertainment games, and pathological affective behavior of NPCs in therapeutic games. Psychological plausibility simply means that the emotions resulting from the model are predictable from a psychological point of view, not that they are "normal" or appropriate to experience at a particular moment in time. Physics engines can model realistic destruction just as easily as realistic movement; the laws of physics don't change and the laws of emotion don't change either. If one assumes that it is difficult to add emotional NPCs to the game design (see pilot results), that emotion simulation means that a full range of affective characteristics must be included (mood, personality, etc.), or that expensive additional game assets are needed for emotions to bring a noticeable difference to the player (pilot study), then one puts up unnecessary walls. Even simple emotions, generated by a rudimentary appraisal engine, can provide interesting game mechanics ([14], and section "Appraisal Engines Enable Novel Gameplays and Genres"). *An appraisal engine for games should hide the complexity of emotion elicitation*, *and provide easy to use functionality and flexible control for simple and complex cases of emotional NPCs*.

Putting up such walls limits the designers' creativity. A telling indication that this occurs is that game developers state that even though *they* see how emotional NPCs could be used in games, players don't demand emotional NPCs (pilot study). For these walls to crumble, different emotion-based gameplays need to be developed and games need to hit the charts. In fact this has happened already with one emotional game genre in the form of the *The Sims* series by Electronic Arts. The Sims' emotions illustrate how emotion models can be used to add realism and fun to the characters in a management/simulation type game. Although successful, this is only one way in which emotions can impact gameplay. Perhaps there is a perceived risk for other developers in adding emotion modeling to NPCs because this means entering the domain of another developer. This risk is real. Apart from the effort to add novel technology that needs to pay off at some point, a company might risk "collateral damage" to its reputation for "trying but failing" to do a better job. To understand the potential of adding emotions to NPCs, experimenting with novel game design and novel gameplays in smaller and simpler games is essential. Developing a computational model of emotion that "runs" is not sufficient. *A successful emotion engine for games should make explicit how the engine facilitates novel gameplay, otherwise the engine does not have perceived value*.

In the remainder of this chapter we discuss how some of the concerns about model-based emotions for NPCs can be addressed. We do this by discussing our own work, because we know this work best. We do not claim we are the only ones trying to address these issues. In our view, these concerns revolve around three aspects of emotion modeling: the need for a black-box, easy to use emotional appraisal engine; the need for easy integration in different NPC architectures; and the need for seeding creativity with novel emotion-enabled game genres. A black-box emotional appraisal model that is easily integrated in different settings is needed to resolve the problem of AI dependency. Novel game genres are needed to explore whether there is a market for games with emotional NPCs. If potential players are not presented with interesting games with novel gameplays, then demand for such games will never arise. Recall that no players asked for Pac-Man either.

## Emotional Appraisal Engines as Plug-in Modules

To facilitate the development of emotional NPCs, an emotional appraisal engine needs to support the development of both simple and complex emotional NPCs, as well as hide the complexity of the emotion generation process, and simulate the emotions in an AI-independent way. In addition it should be high performing and scale efficiently [26]. In other words, the emotion engine should function analogously to a physics engine: it should provide emotion simulation functionality with a clearly defined API that would not depend on the AI used for controlling the NPCs in the game, as this would limit the engine's capacity to operate in a wide variety of games. To demonstrate that this is feasible, we have developed an emotion appraisal engine, GAMYGDALA [15]. Below we briefly describe how the basic assumptions of this appraisal engine help achieve the aforementioned goals.

GAMYGDALA (available in Java, Javascript, and C#) provides three main functions: *emotion generation* via *cognitive appraisal* based on the OCC model; *dynamic relationships among NPCs*, based on a simple like/dislike scheme; and NPC *affective dynamics modeling* including integration of emotional appraisal over time, intensity and decay, and a translation of the categorical emotional state to a the dimensional abstraction defined by the dimensions of Pleasure, Arousal, Dominance (PAD).

To use GAMYGDALA, a game developer first defines the NPC's set of goals (these can include achievement, maintenance and avoidance goals, and can change over time). The developer then decides how particular game events impact these goals, and sends these annotated events to the appraisal engine for processing. These specifications are sufficient to elicit emotions in the NPC. In addition, the developer can configure relationships among NPCs (relationships also develop as a result of emotional appraisal). The NPC relationships then form the basis for social emotions such as feeling resentment (negative relation towards another and a positive event happening to that other) or feeling happy-for another agent (positive relation and positive event). Goals and events need to be specified only for the event appraisal necessary for emotion generation, and need *not* be related to the game AI.

GAMYGDALA is game-AI independent because it only performs a "black-box" appraisal function. It defines the minimum interface needed to implement (a subset) of the OCC model and addresses a baseline level of affective dynamics. To demonstrate GAMYGDALA's AI independent nature, we discuss in section "Integrating Emotional Appraisal Engines with NPC Control" its integration with a cognitive agent programming language [27], a system for defining semantic game worlds [28], a narrative generation engine [29], and with *Phaser* (a Javascript game engine). In all cases the same interface was used based on the annotation of events and the definition of agent goals, as outlined above.

It is possible to simulate emotions for a diverse range of NPCs, especially simpler cases of NPCs, where the set of goals for one NPC is limited [14]. More complex cases, such as when an agent has a large set of goals that include achievement and avoidance goals, when it is important that the agent interpret the event in relation to previous events, or when an event acts on two different goals in opposing directions (e.g., where one is an achievement and the other an avoidance goal), need to managed by the game developer. "Managed" in this case means that the developer needs to decide which event to appraise in relation to which goals, because GAMYGDALA is simply a black-box appraisal engine, and does not implement any reasoning, memory or attentional processing itself. While this limits GAMYGDALA's functionality, implementing these functionalities in an appraisal engine would necessitate making assumptions about game AI and NPC behavior generation, and would violate GAMYGDALA's objective of maintaining independence from the existing NPC AI architecture.

## Integrating Emotional Appraisal Engines with NPC Control

In this section we demonstrate the notion of emotional appraisal engine independence from the NPC AI by illustrating how an emotional appraisal engine can be used with widely different ways of controlling NPC behavior. In particular, we show how we used GAMYGDALA to model the emotions of NPCs developed with the *Phaser* game engine framework, with the cognitive agent programming language GOAL [27], a system for generating semantic worlds and crowds called Entika [28], and with narrative generation [29]. This section thus serves as proof for the claim that black-box appraisal is an approach that enables reusability of the emotional appraisal functionality and can encapsulate emotion modeling complexity.

## *Simulating Emotions in Semantic Worlds*

Creating a virtual world where interaction with almost every object is possible poses many difficult challenges that are far from being solved, including the complexity of maintaining all possible interactions with, and among, the game entities [30]. In particular, to enable the objects with which the player interacts to have an emotional impact on an NPC requires defining some meaning, or *semantics*, for the objects and for those interactions. One approach aimed at solving this problem is *semantic game worlds* [28]. These worlds are designed by defining and choosing the entities populating them from among specific classes, each with their own unique properties, such as attributes, roles and services [31]. With this approach, the objects within the game world themselves carry the information of what they are actually useful for, or able to. It is therefore possible for NPCs to query the game world and discover usable objects for their purposes. In order to facilitate the creation of semantic worlds, a framework called *Entika* was developed [28], that supports a simple and intuitive definition of semantics, promotes re-usability and facilitates object behavior customization. More recently, this framework has been applied to the specification and simulation of the motion behavior of a crowd of agents. For this, a semantic crowd editor was developed aimed at defining crowd templates in a portable way, allowing their reuse for virtually any environment in which the available objects are spontaneously used by other agents in a meaningful manner. This is achieved by having each agent query the environment in order to find whatever objects are deemed suitable to fulfill its goals [32].

With this basis, implementing NPC emotions in a game world involves supporting and integrating two essential elements: semantics and emotions. The semantics component involves creating a semantic game world, populated with entities (living or otherwise) with clearly defined semantics. The emotions component involves analyzing how (un-)desirable the interaction with objects an actors is for the goals of each character, and generating the corresponding emotion(s) accordingly. Semantics in the context of emotion simulation can be rephrased as *the information needed to perform emotional appraisal*. This is the approach behind the integration of Entika and GAMYGDALA.

Goals are instrumental to the simulation of emotion, and, because every NPC in a game should have some goal(s), defining goals for an NPC is a natural feature of a game world with living entities. Further, whatever happens to the NPC can be seen as an *event*, which consists of an action performed at a given moment in time, involving one or more entities. In addition, events can be associated with several other attributes, so that it becomes possible to specify the goals influenced by the event, as well as to indicate whether this influence is positive or negative. The integration boils down to the following: Entika is used to specify events and NPC goals, and GAMYGDALA processes these events and appraises them in the relation of NPC goals at runtime. The flexibility of Entika is leveraged to defining semantic game worlds, while the emotional appraisal functionality is provided by GAMYGDALA to emotionally interpret what happens in the world.

For the configuration of the appraisal engine, various parameters are involved that are provided by Entika such as *belief likelihood*, modeled as the credibility of an event happening; *utility*, specified for each goal of an NPC, indicating the degree (positive or negative) to which the entity wants a given goal to be fulfilled; and *goal compatibility*, i.e., the extent (either positive or negative) to which an event influences the likelihood of a goal being achieved.

The coupling between Entika and GAMYGDALA demonstrates two concepts. First, that appraisal can be easily embedded in a semantic world. For this, abstractions that are relevant to an agent's emotional state are very naturally developed using semantics. Notions such as goals, beliefs and agents can be semantically tied to one another in a way that is generic, scalable, intuitive and independent of implementation platform. Second, that it is feasible to define virtual environments based on semantics, which yield plausible emotional states for NPCs by using a generic emotional appraisal engine. For the full report of this study see [33].

## *Simulating Emotions in Phaser, a Javascript Game Engine*

*Phaser* is a Javascript-based game engine. A Javascript plug-in for Phaser has been developed as a wrapper around GAMYGDALA's public API. Using this plug-in one can configure the appraisal engine to appraise events in the following way. For every NPC that needs emotions, the game developers generate an agent entity in the appraisal engine. The developer then creates goals for each agent, including a goal utility ([-1, 1]). Finally, if needed, the developer creates a positive or negative relationship ([1–, 1]) between NPCs, using a single *createRelation* method. When an event needs appraising, the developer calls the *appraise* method, and defines how the event impacts particular goals. GAMYGDALA then determines which agents are affected and what emotions result from the appraisal, as well as how the NPCs' relationships are affected. This simple API allows the generation of emotional NPCs in a way that shields the developer from the complexity of appraisal itself, and is another example of how black-box appraisal can facilitate the integration of emotions in a different type of NPC development environment. Of course, the integration approach is the same as for the integration with Entika: define agents, define goals and define how the events impact the goals.

## *Simulating Emotions in Cognitive Agent Programming*

A very different setting for the controlling of NPC behavior is cognitive agent programming. Here an agent (or NPC) is controlled using cognitive reasoning in the form of rules that trigger actions based on preconditions that need to be met in order for the rules to fire. In the agent programming language GOAL [27], such agent behavior is specified by a programmer who defines rules, goals and domain knowledge. An agent has a current mental state, representing the state of affairs, and the agent reasons using a rule set about what to do next based on this current mental state. When a rule fires, it triggers an action. Examples of actions are movements, firing bullets, or anything else that can be implemented by the physical embodiment of the agent in its environment. GOAL has been used to control game bots in, for example, Unreal Tournament [34]. GAMYGDALA has been added to GOAL in two different ways: as an integration in the reasoning cycle, and, as a plug-in emotional appraisal module. Here we only explain the integration as a plug-in.

The appraisal engine GOAL plugin works in a similar manner as the integration with Entika. At the creation of a GOAL agent instance (after launching a world with its agents), GOAL launches the plug-in and creates a GAMYGDALA agent instance. At design time, the agent programmer defines GOAL program rules in which the appraisal plug-in is instructed to add and remove goals, as well as appraise events. These instructions are implemented as built-in GOAL actions with similar arguments as described for the Entika and Phaser integration above. These actions call the appraisal's plug-in functionality, and after each appraisal the resulting emotions are added to the agent's mental state as belief predicates, e.g., *emotion*(*happy*, *0.8*). Again, this integration takes place in a very different development environment, but the integration approach is the same.

### *Simulating Emotions in Narrative Generation*

A different approach to integrate appraisal was used in the affective storyteller, a system that generates stories with simulated actor emotions based on the events that happen to the actors in the story [29]. In this system, GAMYGDALA is integrated into the story generation process. The story generation is driven by actors taking actions based on their goals and the current state of the story. Actors and goals are also configured in the appraisal engine, and whenever an event occurs in the story it is automatically annotated and sent to the appraisal engine for processing. The automatic annotation is possible because event likelihood and goal congruence are derived from the story at runtime. Goals are annotated with a utility at design time. This means that once the story domain has been set and the actors are configured, the appraisal engine will fully automatically generate emotions. This is a different example of an integration, where the integration is more closely coupled to the AI than in the previous examples. Here, the appraisal engine gets all of its information, except the goal utilities, from the AI that generates the story.

## Appraisal Engines Enable Novel Gameplays and Genres

The discussion above outlined the motivation for, and benefits of, augmenting NPCs with more sophisticated models of emotions and providing a specialized emotional appraisal engine to facilitate this modeling. The appraisal engine discussed here focuses on a subset of the emotion modeling: emotion generation via cognitive appraisal. (A full-fledged emotion engine would also provide the tools to facilitate incorporating the effects of emotions on the NPCs behavior and internal information processing, including perception, decision-making and planning.)

Below we discuss how even relatively simple emotional appraisal engines, such as those discussed in this chapter, can enhance gameplay. We also highlight novel gameplays, even novel game genres, made possible by dynamic models of affective processing in NPCs. We provide examples of such novel gameplays for several of the more popular game genres, and highlight in particular the benefits of explicit emotion modeling in NPCs in serious games.

### *Action-Adventure Games (e.g., Legend of Zelda, The Witcher)*

These games involve a combination of exploration and puzzle/problem solving, where the puzzles/problems range in complexity from simple, concrete tasks (unlock door to retrieve object) to complex interactions with NPCs. Dynamic generation of even the basic emotions in the NPCs would enable them to display more variability in behavior, thereby providing both increasing affective realism, and "surprises" during the gameplay. Models of more complex social emotions would also provide the opportunity for creating sophisticated 'social puzzles', where the player's progress in the game would necessitate inducing a particular emotion in the NPC, thus shifting the realm of the puzzles from manipulating physical objects to creating and managing complex social interactions. Games like *Crusader Kings II* already take up such an approach, where management of relationships is an important aspect in growing the player's medieval empire.

What an appraisal engine can bring to this is the straightforward way of modeling how NPCs react to attempts to manipulate their emotional state. This would allow the game developers to embed this enhancement in the gameplay, even in games where emotion is not a key focus. It would also allow the generation of "emotional puzzles" for example as a proper mini game. See [14] for an implemented example of such an emotional puzzle game.

### *Fighting and First-Person Shooter Games (e.g., Mortal Combat, Doom)*

Both of these genres focus on direct combat with and/or killing of the opponent. The player typically views the opponent directly, although some games provide a third-person view of the gameplay. The affective realism of the NPCs in this genre is limited to rudimentary expressions of aggression, and, less frequently, fear. In existing games these emotions are typically scripted, and little variability or nuanced NPC behavior is possible. Games augmented with model-driven emotion generation would enable the NPCs to vary their emotional reactions to the evolving context, both in terms of which emotion is displayed when, and the intensity of that emotion; again, providing less predictable behavior and thus more engaging gameplay.

While the NPCs' display of aggression and rudimentary fear may be adequate for most FPS players, one can imagine an interesting evolution of this genre, where the opponent NPCs could display more complex dynamics of these emotions, and include additional emotions, such as sadness, guilt, envy, happiness or pride. Display of this broader set of emotions would not only enrich the player experience, but could also 'humanize' the typical FPS gameplay. Also, such social emotions could impact NPC capabilities later in the game (e.g., guilty NPCs fight less aggressively or adopt different tactics). While this might not be the experience sought out by most current FPS players, such augmentation could be a welcome development in this genre and might mitigate the violence de-sensitization effects that are attributed to FPS games.

In general, NPC opponents could adopt different strategies depending on their emotional state. For example, a fearful NPC opponent that is immune to particular player moves, because fear makes the NPC react more quickly and move in a more random manner. Beating the fearful NPC would mean chasing it until exhaustion, while beating an angry NPC could be done with a few well-placed special blows. The player could manipulate the emotional state of the NPC before or during the combat scene, by manipulating the game objects or via interaction with other NPCs. Such gameplay would be easily supported by an appraisal engine and would require relatively simple additional game assets to display the NPC emotional state to the player.

### *Real-Time Strategy Games (RTS) (e.g., Age of Empires, Warcraft)*

RTS games involve a competition with another player or NPC for some set of resources. It is easy to see how increased affective sophistication of the NPCs would enhance the gameplay. As in the action adventure games above, not only would the element of surprise, enabled by model-driven emotion generation, provide a more engaging experience, but the possibility of inducing distinct emotions in the NPCs to motivate them to engage, or not engage, in specific behavior would create an entirely distinct domain for player behavior: the social domain. In contrast to the current RTS physical domains, where physical objects or resources are manipulated, the addition of a social domain, enabled by the NPCs affective realism, would allow the players to 'manipulate' and control social resources, such as the NPCs' good or ill will. This capability would create much more believable gameplays, reflecting more accurately the reality of political and military strategies this genre aims to emulate. This is in line with, for example, *Crusader Kings II* (a turn-based strategy game), where relationships play an important role in how the political and military struggles unfold over the different generations.

### *Role-Playing Games (RPGs) (e.g., Baldur's Gate, Skyrim)*

RPGs are fantasy-adventure games where the player(s) engage in a complex series of tasks to achieve the ultimate game goal (e.g., save humanity from a plague, defeat an evil cult, etc.). Novel gameplays enabled by emotion modeling in NPCs include those outlined above for the real-time strategy games. However, the realm of social interaction and the possibility of achieving affective and social tasks is even more important in RPGs, since the player has the opportunity to develop deep interactions with the NPCs over the course of playing through different quests [5]. In addition, a variety of emotional state manipulation can be implemented and quests can be (de)activated based on the emotion of particular NPCs, thus making emotional manipulation a key element of the quest progression. The same applies to simulation games such as The Sims.

### *Arcade and Platform Games (Pac-Man, Mario Bros)*

Even basic emotion simulation can enhance arcade-like gameplay. Behavior of "baddies" in the game world can be made dependent on their emotional state. This is analogous to the influence of emotions on NPCs in the FPS and Fighting genre. For example, emotion-augmented baddies might grow to hate or like you depending on what items you gather in a level. Levels could be built around the idea that the player has to manipulate the emotions of NPCs in such a way that certain baddies become a threat while others do not. An example of such gameplay is presented in [14] in the form of a game called *friend or foe*.

### *Serious Games*

Currently, games are used primarily for entertainment purposes. Increasingly however, games are being adapted for instructional, training and therapeutic purposes. In fact, serious gaming is the fastest growing segment of the game industry. All of the above genres can be adapted to serious gaming and the increasing affective sophistication of the NPCs, facilitated by affective game engines, is even more relevant in serious gaming contexts, particularly in therapeutic serious games, including games used to augment psychotherapy.

The objective of serious games in psychotherapy is to support and augment face-to-face therapy, by providing opportunities to both experience problematic situations and to practice new behavior and coping strategies [35, 36]. The immersive quality of games, particularly games that use elements of virtual reality (e.g., head-mounted displays) provide a unique means of creating customized physical and social situations that induce undesirable behavior, and the opportunities to develop, *in* *virtuo*, strategies to cope with these situations. The simulation of psychologically plausible emotions is especially relevant for simulation of complex social situations.

An emotion game engine that would support the development of these functionalities in a manner that is verified with respect to psychological plausibility would be especially useful for developing serious games, because it would enable the game designers and developers to focus on the core of the training or treatment, while the implementation of the psychologically plausible NPC emotions would be provided by the emotion simulation module.

## Final Remarks

In this chapter we have motivated the need for specialized emotional appraisal engines for games, to facilitate the modeling of dynamic and more complex affective behavior in NPCs. We focused on modeling emotion generation via cognitive appraisal and presented evidence that when appraisal is approached in a black box fashion, where the appraisal process is decoupled from the logic controlling the NPCs, it is possible to embed the appraisal process as a plug-in in many different NPC architectures. This chapter is limited to a discussion of emotion generation models, and focused on discussion of approaches that would facilitate the development of model-driven emotions for NPCs. The effects of emotions on the NPCs internal processing and behavioral choices, also part of emotion modeling, have not been addressed.

Currently, the state of the art in emotion modeling literature is not advanced enough to have standard and agreed upon models of how emotion influences agent behavior. (See [37, 38] for a discussion of existing efforts and challenges in this area.) Future emotion engines for games should also include explicit models of the effects of emotions on the NPCs internal information processing (e.g., perception, decision-making, planning), as well as behavior choices. These models could then be used by game developers to bias the NPCs' reactions in the game environment, including the social environments; that is, the NPCs' reactions to the behavior of the player and other NPCs. As is the case with models of emotion generation, here it would also be important to hide complexity of these models. Just as a black-box appraisal plug-in, a black-box emotion effects plug-in should also be independent of the AI used to control the NPC, and this is perhaps an even greater challenge. Existing efforts in this area provide the basis for facilitating this type of modeling in emotion game engines. For example, the MAMID modeling methodology [39, 40] provides a means of encoding the effects of a broad range of factors, including emotions and personality traits, on internal processing. The parameters control the speed, capacity and biasing within individual architecture modules, and facilitate not only the modeling of the distinct effects of emotions on NPC behavior, but also the rapid construction of a wide variety of distinct NPC personalities. These capabilities would provide additional tools for the game developer to facilitate the construction of increasingly affectively realistic NPCs.

Finally, an aspect we have not touched upon is testing and debugging emotional appraisal. Insight into how one can debug and test emotional models is needed to manage the risks involved in having autonomously behaving entities in a game. This is of specific importance in large, procedurally generated worlds, where game designers set constraints for the game generation but do not have full control over how the game world and its inhabiting NPCs are configured.

Notwithstanding these limitations, we have shown the potential of emotional appraisal engines to transform the development of affective games. Model-based emotion generation for NPC can take the use of emotions in game design to the next level, and can be a basis for many novel gameplays.

## References

1. Schell J (2008) The art of game design: a book of lenses. Morgan Kaufman Publishers, Amsterdam
2. Karpouzis K, Paiva A, Isbister K, Yannakakis GN (2014) Guest editorial: emotion in games. IEEE Trans Affect Comput 5(1):1–2
3. Yannakakis GN, Paiva A (2014) Emotion in games. In: Calvo RA, D'Mello S, Gratch J, Kappas A (eds) Handbook on affective computing. Oxford University Press, New York, pp 459–471
4. Hudlicka E, Broekens J (2009) Foundations for modelling emotions in game characters: modelling emotion effects on cognition. In: Affective computing and intelligent interaction and workshops, 2009. 3rd international conference on ACII 2009, pp 1–6
5. Freeman D (2004) Creating emotion in games: the craft and art of emotioneering™. Comput Entertain (CIE) 2(3):15–15
6. Hudlicka E (2008) Affective computing for game design. In: Proceedings of the 4th Intl. North American Conference on Intelligent Games and Simulation, pp 5–12
7. Hudlicka E (2003) To feel or not to feel: the role of affect in human-computer interaction. Int J Hum-Comput Stud 59(1–2):1–32
8. Hudlicka E (2008) What are we modeling when we model emotion? In: AAAI Spring symposium 2008
9. Reilly WS, Bates J (1992) Building emotional agents. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA
10. Bidarra R, Schaap R, Goossens K (2010) Growing on the inside: soulful characters for video games. In: Proceedings of the IEEE conference on computational intelligence and games
11. Hudlicka E (2009) Affective game engines: motivation and requirements. In: Proceedings of the 4th international conference on foundations of digital games. ACM, Orlando, pp 299–306
12. McCoy J, Treanor M, Samuel B, Reed AA, Mateas M, Wardrip-Fruin N (2014) Social story worlds with comme il faut. Comput Intell AI Games IEEE Trans on 6(2):97–112
13. McCoy J, Treanor M, Samuel B, Mateas M, Wardrip-Fruin N (2011) Prom Week: social physics as gameplay. In: Proceedings of the 6th international conference on foundations of digital games. ACM
14. Broekens J (2015) Emotion engines for games in practice: two case studies using Gamygdala. In: Gratch J, Schuller B (eds) 2015 international conference on affective computing and intelligent interaction (ACII), Xi'an, pp 790–791
15. Popescu A, Broekens J, Someren MV (2014) GAMYGDALA: an emotion engine for games. IEEE Trans Affect Comput 5(1):32–44
16. Hudlicka E (2011) Guidelines for designing computational models of emotions. Int J Synth Emotions (IJSE) 2(1):26–79

17. Marsella SC, Gratch J (2009) EMA: a process model of appraisal dynamics. Cogn Syst Res 10(1):70–90
18. Aylett R, Vala M, Sequeira P, Paiva A (2007) In: Cavazza M, Donikian S (eds) FearNot! – an emergent narrative approach to virtual dramas for anti-bullying education, in virtual storytelling. Using virtual reality technologies for storytelling. Springer, Berlin, pp 202–205
19. Ochs M, Sabouret N, Corruble V (2009) Simulation of the dynamics of nonplayer characters' emotions and social relations in games. Comput Intell AI Games IEEE Trans on 1(4):281–297
20. Mateas M, Stern A (2003) Façade: an experiment in building a fully-realized interactive drama. In: Game developers conference
21. Ortony A, Clore GL, Collins A (1988) The cognitive structure of emotions. Cambridge University Press, New York
22. Marsella S, Gratch J, Petta P (2010) Computational models of emotion. In: Scherer KR, Bänziger T, Roesch E (eds) A blueprint for affective computing. Oxford University Press, New York, pp 21–45
23. Bartneck C (2002) Integrating the occ model of emotions in embodied characters. In: Proceedings of the workshop on virtual conversational characters, pp 39–48
24. Steunebrink BR, Dastani M,, Meyer J.-JC (2008) A formal model of emotions: integrating qualitative and quatitative aspects. In: Proceedings of the European Confernce on Artifical Intelligence (ECAI '08). IOS Press, pp 256–260
25. Steunebrink BR, Dastani M, Meyer J.-J.C (2007) A Logic of emotions for intelligent agents, In: Proceedings of the 22th national conference on artificial intelligence (AAAI '07). AAAI Press, pp 142–147
26. Broekens J, DeGroot D (2004) Scalable and flexibel appraisal models for virtual agents. In: Mehdi Q, Gough N (ed) Proceedings of the fifth game-on international conference, pp 208–215
27. Hindriks K, Meyer JJ (2009) Toward a programming theory for rational agents. Auton Agents Multi-Agent Syst 19(1):4–29
28. Kessing J, Tutenel T, Bidarra R (2012) Designing semantic game worlds. In: Proceedings of the third workshop on Procedural Content Generation in Games. ACM, Raleigh, pp 1–9
29. Kaptein F, Broekens J (2015) The affective storyteller: using character emotion to influence narrative generation. In: Brinkman W-P, Broekens J, Heylen D (eds) Intelligent Virtual Agents (IVA2015). Springer, pp 331–334
30. Tutenel T, Bidarra R, Smelik RM, Kraker KJD (2008) The role of semantics in games and simulations. Comput Entertain (CIE) 6(4):57
31. Kessing J, Tutenel T, Bidarra R (2009) Services in game worlds: a semantic approach to improve object interaction. In: Natkin S, Dupire J (eds) 8th international conference on entertainment computing (ICEC 2009), Paris, 3–5 Sept 2009. Springer, Berlin/Heidelberg, pp 276–281
32. Kraayenbrink N, Kessing J, Tutenel T, de Haan G, Bidarra R (2014) Semantic crowds. Entertain Comput 5(4):297–312
33. Ercan S, Harel R, Peperkamp J, Yilmaz U (2014) Virtual humans in games: realistic behavior and emotions for non-player characters. BSc thesis, Intelligent Systems, TU Delft, Delft
34. Hindriks K, van Riemsdijk B, Behrens T, Korstanje R, Kraayenbrink N, Pasman W, de Rijk L (2011) Unreal Goal Bots, In: Dignum F (ed) Agents for games and simulations II. Springer, Berlin, pp 1–18
35. Hudlicka E (2016) Virtual affective agents and therapeutic games. In: Luxton DD (ed) Artificial intelligence in behavioral and mental health care. Elsevier
36. Rizzo A, Shilling R, Forbell E, Scherer S, Gratch J, Morency L-P (2016) Autonomous virtual human agents for healthcare information support and clinical interviewing. In: Luxton DD (ed) Artificial intelligence in behavioral and mental health Care. Elsevier
37. Reisenzein R, Hudlicka E, Dastani M, Gratch J, Hindriks K, Lorini E, Meyer JJC (2013) Computational modeling of emotion: toward improving the inter- and intradisciplinary exchange. Affect Comput IEEE Trans 4(3):246–266

38. Hudlicka E (2014) From habits to standards: towards systematic design of emotion models and affective architectures. In: Bosse T et al. (ed) Emotion modeling: towards pragmatic computational models of affective processes. Springer, pp 3–23
39. Hudlicka E (2003) Modeling effects of behavior moderators on performance: evaluation of the MAMID methodology and architecture. In: Proceedings of 12th conference on behavior representation in modeling and simulation, Phoenix
40. Hudlicka E (2008) Modeling the mechanisms of emotion effects on cognition In: Proceedings of the AAAI fall symposium on biologically inspired cognitive architectures. AAAI Press, pp 82–86