An Experimental Analysis of NPC as Affective Agents in Real World-based Games

Rainier Sales, Esteban Clua, Daniel de Oliveira, and Aline Paes
Computer Science Department, Computing Institute
Fluminense Federal University (UFF)
Niterói, Brazil
{rsales, esteban, danielcmo, alinepaes}@ic.uff.br

***Abstract*— Existing games are based on the gameplay for achieving the necessary fun factor, emotive elements are becoming common and in some cases even necessary when modeling Non-player character (NPC) behaviors. In this way, including and bringing emotional aspects from the real world may make the game closer to the reality, turning the virtual characters more like humans. This paper proposes an implementation of NPCs as affective agents by including personality, mood and emotive features based on well-established psychology models. Our affective agent model is implemented as a NPC in a memory game, where the different emotive aspects are introduced in order to analyze the variability of its behavior based on the emotions. The results of 2,200 gameplay sessions using affective agents and considering different emotions show that the agent behavior oscillates between a perfect agent and a random agent. We can observe a behavior more similar to human behavior, where the performance of their actions became directly linked to your current emotional state.**

Keywords— Affective Agents, Artificial Intelligence.

I. INTRODUCTION

Emotions play a key role in the user experience, both in entertainment games and in serious games developed for education, training, assessment, therapy or rehabilitation. The gaming community has recently recognized the importance of emotion in the development of more engaging games, and the area of affective gaming is receiving increasing attention [1]. Since the real world is a source of emotions, game developers try to consider external events during the development of the game. However, these events are not updated after the game release. The game is only updated either to improve playability or to fix bugs. However, the game should become more attractive to the player if real world events that are capable of interfering in the game could be captured and produce consequences in the game session. Let us illustrate this scenario with the following real example: in the game Top Spin 4™, the top two tennis players that are difficult to defeat are Rafael Nadal and Novak Djokovic. This is comprehensible since when the game was developed these were world number one and number two respectively. However, during the year 2012 Nadal was injured and did not play many tournaments. However, in the game, its character was still very difficult to beat although the real Nadal was injured. If the game could reflect the real world, Nadal should present a worse performance due to the knee injury. This way, the playability and immersion could be improved.

In this paper we present a strategy to build convincing affective NPC based on human psychological aspects. In order to be objective and present results, we have developed a memory game, where we can directly analyze the gains and differences in the use of affective agents in digital games that model the real world. Even though the simplicity of this type of game, the results reinforced that this approach could be applied in more complex games. Besides the main goal of the paper that is a case study on affective NPC, we highlight another specific objective that is the validation of theoretical models of psychology by means of a computer-based implementation. These contributions are relevant as they aim at improving and maximizing the sense of reality of the player with the game, and consequently increase the sense of immersion and fun.

This paper is based on the following structure: Section II brings related work. Section III presents the psychological aspects of characters as agents affective, describing the model of personality, emotion, mood and response. In section IV we present the architecture developed and used in the tests. In section V we present the simulator PMESAA used to implement the case study. In Section VI we present the case study. In section VII we conclude the paper.

II. RELATED WORK

There are some work related to affective NPC that are worth mentioning, although none of the work presented here have the same purpose of the approach presented in this paper. Actually, they do not use affective agents in games, but in robotics. The past 15 years have witnessed a rapid growth in computational models of emotion and affective architectures. Researchers in cognitive science, artificial intelligence and human computer interaction (HCI) proposed models of emotion for theoretical research regarding the nature of emotion, as well as a range of applied purposes as to create more believable and effective synthetic characters and robots, and to enhance HCI [2].

Kshirsagar and Magnenat-Thalmann [3] developed a model of multilayer agents, aiming at simulating humans' personality, mood and emotions. There, the authors implemented the Big Five model, but, they have considered a Bayesian network to select the agent behavior. The main point that differentiates the work from Kshiragar and Magnenat-Thalmann from the proposed in this paper is the type of application. In that work, affective states do not modify the performance of the NPC.

The architecture named ALMA, proposed by Gebhard [4], is one of the main references in the implementation of emotional agents. The architecture is implemented through three emotional states: emotion, mood and personality (the same models are adopted in this paper). The architecture is used in a project named Virtual Human, which combines state of the art in computer graphics technology and dialogue generation. In this work, the emotional states of the agent are used to modify the facial expression, selection of words and phrases, selection of strategies etc.

The architecture developed by Kasap *et al.* [5] presents a model of emotion-based memory to build long-term relationships between users and virtual characters. Such a relationship is the accumulated result of the evaluation (positive and negative) that the agent gathers during their interaction. This evaluation is made by means of emotions associated with perceptions. To develop the architecture they used the model of the Big Five personality factors, the model PAD Mood [6] and OCC model [7] of emotion, the same as we used. However, in that work, emotions are used to create a bond between the NPC and the player. We follow a different approach, as we use the emotions to change the performance of the NPC during the game.

III. Representation and Modeling of Psychological Aspects with Affective Agents

The use of affective NPC has gained importance in the digital games area, motivated by the need to represent humans in a more realistically way. Many of the current games do not offer support to affective agents. Existing games do not answer questions such as "how to respond to a particular event of the real world?" and "how long an event is perceived?" These problems are not solved even if the developer tries to map all possible events in the game, because the number of combinations is huge. The affective NPC can make decisions autonomously based on events occurring during game play. To make a closer representation of the reality, a NPC should be based on a model of emotions derived from psychology [8] [7]. The first step in creating an affective NPC is defining your personality, since the personality of a human being is basically what differentiates it from other psychologically.

The model is the most widely used personality known as "The Big Five Model" [9]. This model describes and classifies the human personality into five factors (Neuroticism, Extraversion, and Openness to Experience, Friendliness and Responsibility). In this model, an individual's personality is built over time, being modified during their lifetime, hence the importance of the historical information. The mood, on the other hand, it is a more immediately being changed constantly. A representation often used to represent the mood is Model PAD (Pleasure, Arousal and Dominance) [6]. Beyond personality and mood, we also model the emotion of the NPC. Emotion is a psychological response to an event to an individual, in order to change the current mood state. The emotion model considers internal and external events and is essential for proper communication between the NPC and the events surrounding it. The model known as OCC [7] uses a set of mapping rules, and its proven relevance in the area of psychology and neuroscience. For each NPC, is initially defined personality whose values vary in the range [-1, 1], where they are mapped the five factors of the Big Five model. Based on the defined personality we generate base mood, which is the mood state that represents the NPC without interference events. To calculate the base mood is used the expression defined by Mehrabian [6] where the attribute values are personality transformed into a point in the state-dimensional model PAD. When an event occurs, this comes accompanied by intensity. An event perceived by the NPC may vary in type. It can be a death or a goal (as in the example presented in the introduction).

The association of the emotion of the event agent is made by a vector sum (the attributes of models). From this sum, we obtain the real change in agent. Thus, through the fuzzy logic; the eight mood states are mapped from 22 states of emotion agent, to represent the change in the performance of the NPC in response to some event. From the final mapping, we can define the influence of mood (positive or negative) and its intensity acting directly on the attributes of the NPC. Thus, it is essential to capture the information personality, mood and emotion so that developers can track the history of this information after a play session or an NPC to develop more sophisticated.

IV. Affective agent Architecture

In this section we describe the architecture for Affective agents applied to the game used as case study in this paper. For more details about the architecture please refer to Sales *et al.* [10]. The architecture was implemented to be coupled to any game that is plausible using affective NPC, without major modifications. The idea is to implement cartridges for each different game and the architecture components integrate with these cartridges. The architecture is composed by four modules: Creation module, Perception module, Organism module and Response module, as shown in Figure 1.

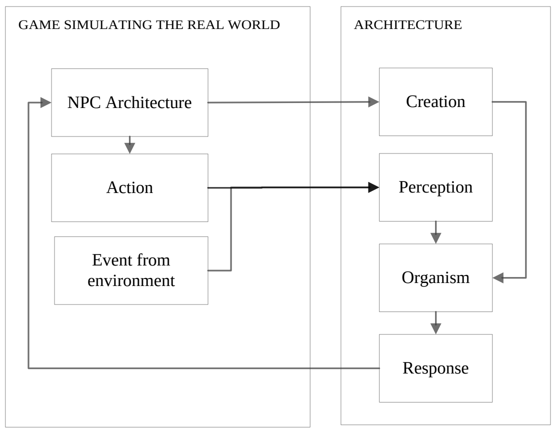

Fig. 1. Architecture and game communication

In the architecture, a NPC performs actions in the game, and these actions can be mapped as events between agents or events not controlled by any NPC (*i.e.* environmental events). Using the architecture, the NPC communicates with the creation module, when an event happens in the game. The architecture internally performs the calculations in the organism module and then provides a response using the response module, which in turn performs an action associated to the NPC. The creation and perception module of events are responsible for initiating and generate all the necessary features for architecture. It is initially defined personality in a vector P varying between [-1, 1], which are mapped to the five factors of the Big Five model. With defined personality, the architecture generates base mood. This mood represents the NPC without interference events. To calculate the base mood we used the formula defined by Gebhard [4] where the vector character is transformed into a point in three-dimensional state of mood.

In the next step, the architecture invokes the organism module. Before mapping the events, it is necessary to map the agent's current emotion. Using the base mood it is possible to calculate the emotion intensity from the table of direct mapping defined by Mehrabian [6] and presented in Table I. With the entire configuration required to run the affective agent in hands, we can insert events. An event perceived by the agent can be of a variety of types such as a death or a victory. As the event and the emotion of the NPC uses the same model, the association of the event to the agent is performed by a vector sum.

The next step, in response module, is where we have the real change in the agent. As an emotion is a temporary state, we have to map it from the direct conversion of table defined from Mehrabian [6] in the same way. This way, we map emotion to current mood. At this point we change the current mood due to the insertion and interpretation of one event by the NPC. In the final step we use fuzzy logic to map the 22 mood states to the three states of emotions, in order to represent the change in the performance of the NPC, as show in the Table I. This table links the emotion components to the cognitive components defined by Picard [11] (*e.g.* Memory, perception, decision making, learning, motivation, attention, prioritization, planning and creativity). In short, the end result generates the value to be changed by the performance of the NPC between [1,-1].

This value can be used in two forms to change the performance of the NPC in the game: (1) using a simple rule of three and (2) using a local the rule of three. The direct rule of three considers the default value of the agent as the maximum value (100%), and thus corrects the value of the agent from [-1, 1] to [0, 1], hence the events interfere only negatively on the NPC. The local rule of three however, does not correct the value and maps the values considering a range of interference, so the default agent is associated with an intermediate value in this range, and consequently the events interfere positively or negatively to the NPC, as show in the Table I.

TABLE I. MAPPING EMOTION AND MOOD FOR THE PERFORMANCE

Exuberant			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Gratification	0,95	1	0,975	1
Pride	0,9	0,95	0,95	0,975
Happy-For	0,85	0,9	0,925	0,95
Joy	0,8	0,85	0,9	0,925
Love	0,75	0,8	0,875	0,9
Dependent			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Admiration	0,67	0,75	0,835	0,875
Gratitude	0,59	0,67	0,795	0,835
Hope	0,5	0,59	0,75	0,795
Loose			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Satisfaction	0,38	0,5	0,69	0,75
Relief	0,25	0,38	0,625	0,69
Hostile			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Hate	0,13	0,25	0,565	0,625
Anger	0	0,13	0,5	0,565
Docile			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Gloating	-0,25	0	0,375	0,5
Anxious			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Disappointment	-0,31	-0,25	0,345	0,375
Remorse	-0,37	-0,31	0,315	0,345
Shame	-0,43	-0,37	0,285	0,315
Fear	-0,5	-0,43	0,25	0,285
Arrogant			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Reproach	-0,75	-0,5	0,125	0,25
Bored			**Performance Intensity**	
Values	**Min**	**Max**	**Min**	**Max**
Resentment	-0,81	-0,75	0,095	0,125
Pity	-0,87	-0,81	0,065	0,095
Distress	-0,93	-0,87	0,035	0,065
Fears-confirmed	-1	-0,93	0	0,035

V. THE PMESEAA SIMULATOR

In order to present the use of our affective agent architecture in real games, we developed a simulator named

PMESAA - Personality, Mood and Emotion Simulator to Affective Agents, available for download at: http://pmesaa.codeplex.com. In a significant way, the simulator uses the aforementioned architecture in the form of a class, so the purpose in this section is to present mechanisms to support the architecture effective use. The developed simulator consists of three general components described below: (i) **Agent Definition:** it is where we define the settings of the Affective Agent. Initially, it is necessary to insert values for the personality parameters. Then, from the personality model, the simulator computes in real time the base mood of the agent. Next, from the base mood, the simulator computes the current emotion of the agent. (ii) **Events Processing:** Events are actions which have emotional characteristics. A particular action is relevant to the agent when it has an associated level of emotion. An event can be classified by the agent as either an internal event or and external event. All internal events are going to be perceived by the agent. On the other hand, an external event is noticed only when the intensity of the emotion is greater than the value of minimum intensity perception parameter. When an event is perceived by the agent, we have to change its current emotion parameters. In this way, the emotion value assigned to the agent is no longer derived from the personality and base mood, but it also takes into account the external events. (iii) **Agent Behavior Definition:** The agent behavior is defined in terms of its current mood. Initially, this is the same as the base mood. When an event happens, the agent current mood changes, as the simulator computes the new mood of the agent based on the emotions related to that event. When an event is inserted, the performance intensity attributes also changes, to ensure that the emotions are changing the performance of the agent.

A. Example of Application

There are several games that can make use of the architecture proposed here. Among these we can highlight PES 2013™, NBA2K13™, UFC UNDISPUTED 3™, NHL 13™, where the events can be represented by real events that happened in the real world. The simulator proposes modifications contained in the attributes of the NPC during the game. As far as we know, there is nothing similar in games, as in the current architecture NPCs performance is fixed or varying exclusively temporally (physical tiredness, fatigue etc.). These modifications are represented in a chart, which is constantly modified during the game. To modify the attributes of the game we used two different heuristics, the first three named rule of three basically applies a simple rule of three in the default value, considering the value in the database as the maximum value, so the new value of the attribute is given by the proportion of new value performance. Despite being simple and easy to implement, the rule of three only influences negatively the perception of events from NPC. The second rule nevertheless named local rule three solves this problem by performing only the rule of three in a band of values, so the local rule of three is not directly influence the attribute of the agent, but only in a band of value. The local rule of three in this example is considered by most advanced can cause emotions to influence positively and negatively the NPC. Figure 2 represents this chart in a specific instant of the FIFA 2013© game simulation.

In Figure 2, the chart in dark blue (Default) represents the expected performance of the agent when the events are not taken into account. The chart in dark green color (Rule of Three) represents the change in percentage from an isolated event that affects the performance of the agent. Finally, the chart in dark red (Local Rule of Three) shows the performance of the player according to his attributes at the exact moment after noticing an event (in this case the emotion has "shame" with an intensity of 0.9), In this last case, numerically speaking, all attributes obtained a decrease of approximately 10.34% which directly influence the player performance (in a free kick, a penalty or a simple goal kick).

By modifying the performance intensity of the agent from an event, we are going to have a different value to agent attribute. Values of positive emotions are going to be close to the default value of that emotion to an agent, while negative emotions are going to tend to values distant from the default, the approach of classification between positive emotions and negative is coming of the studies of Ortony *et al.* [7].

Fig. 2. graphic Comparison change the attributes of a NPC

B. Validation of the Simulator

To validate the simulation and therefore the operation of the simulator, we have determined commonly acceptable behaviors, based on human reaction to events: **"A Positive input event maximizes positive outputs response"**. In this case, a positive emotion maximizes the positive results. We considered the average results for each attribute in 100 tests for each of the 11 positive emotions, generating a total of 1.100 tests to verify this rule. As we can see in Figure 3, in the results presented, all positive events produced positive results, changing to more than half of the original value. This indicates that the simulator obeys this criterion.

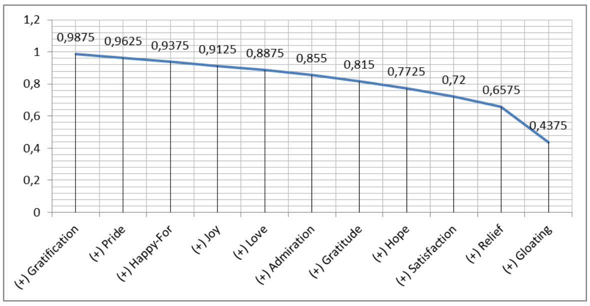

Fig. 3. Positive emotion inputs

"A Negative input event maximizes negative outputs response". In this case, a negative emotion maximizes the negative results. The average results for each attribute (penalties in this case) in 100 tests for each of the 11 negative emotions (generating a total of 1.100 tests) are going to provide the guidance to verify this rule. The results showed in Figure 4 state that all results of negative emotions input values of the attributes of the agents were changed to half the original values. This indicates that the simulator obeys this criterion.

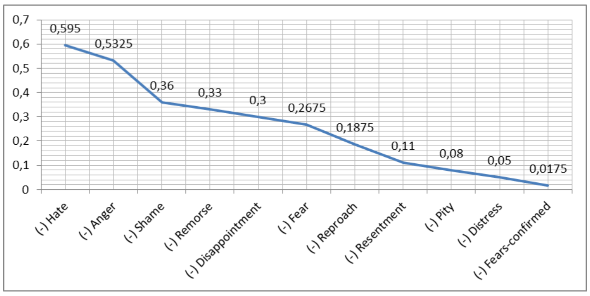

Fig. 4. Negative emotion inputs

"Different intensity of Positive events generates different results of different intensity". To validate this rule, we set the minimum intensity of the attribute performance to 0.5. As the logic applied is the same in all emotions we chose emotion Admiration for example in this test. The results obtained from 100 tests were conducted for each of the three intensities (0.00, 0.50 and 1.00) of the emotion, generating a total of 300 tests and are presented in Figure 5. As the results show, higher intensities tend to positive results and to the maximum performance value. For lower values of intensity, which are still above the minimum value, intensity measures are going to have a more distant performance value of the maximum value of the agent. This indicates that the simulator obeys this criterion.

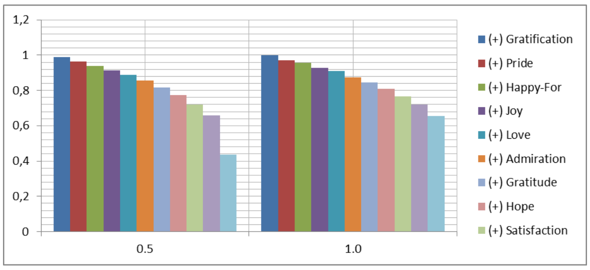

Fig. 5. Different positive emotion inputs

"Entries intensity of negative events generates different results of different intensity". To validate this we set the performance minimum intensity attribute to 0.5. As the logic applied is the same in all emotions we chose emotion "Anger" for example. The results were obtained from 100 tests, conducted for each of the three intensities adopted in the emotion, yielding a total of 300 tests. The results are presented in Figure 6. As the results show, higher intensities have larger negative results, tending in the opposite direction. For lower values of intensity, but above the minimum value intensity, values indicate performance measures farther from the opposite value of the maximum value of the agent.

"Events of lower intensity than the minimum intensity of the agent perception are not even taken into account by the agent". To verify this rule, we use the minimum intensity attribute performance equal to 0.5. As the logic applied is the same in all emotions we choose two emotions, "Admiration" and "Anger". Approximately 100 tests were performed on each of these emotions, generating a total of 200 tests. As the results show, in cases where the intensities are lower than the minimum intensity, the emotion of the event has not even been perceived by the NPC.

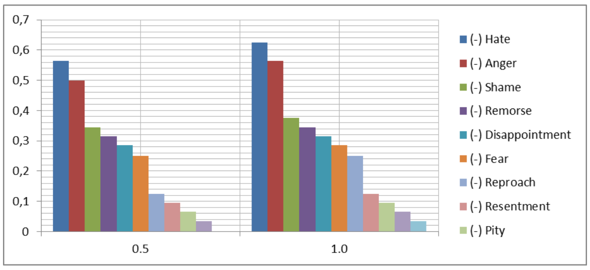

Fig. 6. Different negative emotion inputs

VI. CASE STUDY: THE MEMORY GAME

The case study aims at demonstrating an application of the research presented in a real scenario, and not a simulation. We call this memory game as "Memory Game Test". The game follows the basic rules implemented in a memory game (player vs. computer), so the same rules were implemented in this case study. The human player basically chooses which type of opponent (1). The level of the opponent's memory is updated according to the opponent's choice (2), according to the level of memory and the type of opponent the game sets the facial expression of opponent (3). After configuring the game, the human player can use one of the numbered (4, 5, and 6) buttons to start and end of the game. After starting the game the player begin choosing the cards to view. By clicking on one of the cards (7) it should look the same for the earlier card already opened. If the player chooses right he/she scores a point and (8) continues playing, otherwise the player returns the cards to the same position (7) with the face down, as shown in the Figure 7.

Fig. 7. Experiment 1: Memory Game Test Interface

A. *Experimental Methodology*

In Memory Game Test were implemented three types of agents with different features (I) a random agent, (ii) a perfect agent and (iii) an Affective agent. Below we examine the each type of agent in the memory game context. **Random Agent:** This agent chooses the cards and does not record anything in memory. Thus, the only decision taken by such an agent is to selected randomly two available cards at each time it is its turn to play. Figure 8 presents the flow of the activities performed by the random agent.

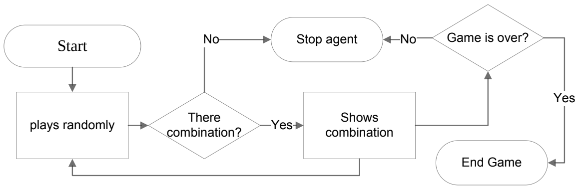

Fig. 8. Random agent flow of activities

A Perfect Agent: This agent is the opposite of the random agent, as it stores as much information as possible concerning which cards were opened for it and for the player, as well as all combinations of cards already opened. Figure 9 presents the flow of the activities performed by the perfect agent.

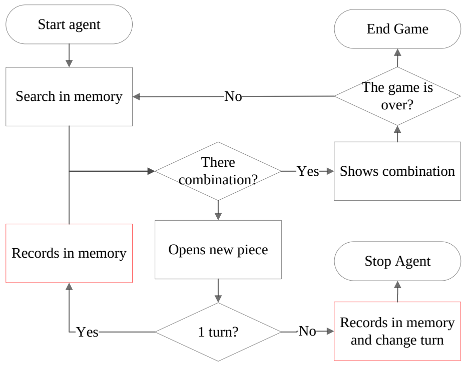

Fig. 9. Perfect agent flow of activities

Affective Agent: the perfect agent may not present the necessary fun factor for a human being, since it should be very hard to be beaten. Thus, we have modified the perfect agent to it look more like humans, including the emotions related to this type of game. Thus, whenever a game card is opened, there is a probability directly proportional to the performance of the agent that it is going to store the information related to that turn. Figure 10 presents the flow of the activities performed by the affective agent.

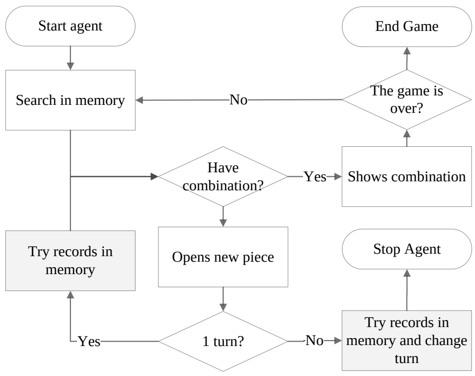

Fig. 10. Affective agent flow of activities

The choice of the memory game was not a coincidence. As Picard [11] divides the components of emotion on physiological and cognitive and the architecture specifically addresses the cognitive components (memory, perception, decision making, learning objectives, motivation, attention, priority planning and creativity). This way the cognitive components are affected as following: (i) Memory: when the agent search for any combination of cards that have been opened, (ii) Perception: when the agent writes to memory the cards that have already been opened by the adversary, (iii) Planning: when the agent seeks a card in memory after opening a new card and (iv) Attention: when the agent writes to memory any card it has already opened. As described by the architecture [10] more complex games tend to use a larger number of components, specifically within this game we used 4 of the 10 components provided by the architecture. We expect that the affective agent can be an intermediate between the perfect agent and the random agent. The affective agent resembles human behavior by simulating failures related to its emotions. On the other hand, the perfect agent never fails and the random provides an unpredictable behavior. The affective agent varies between perfect accuracy and error, since it is based in an event directly linked to the simulated emotion. Next, we present the results and limitations when including an affective agent in the memory game.

B. *Evaluation Criteria*

In order to validate the use of affective NPC in the memory game, we observed the variability in the agent behavior over time. Variability of the behavior is a key variable in games when the logical behavior becomes predictable, as we have a good chance that the player stops playing the game, or even finds logical ways to cheat it. Basically, to validate the range of the variability, it can be used as the time spent on a game criterion or the amount of turns until a match is terminated. The better is the agent, the less is the time spent on a game, while the worse the agent is, more time it will spend in a match. However, when a human is playing against a NPC, the time spent on a game is directly influenced by both the agent performance and the human player performance. Thereby, we have to remove the human bias in the experiment to evaluate the NPC. To achieve this, we defined that the human player always hold the last move made by the NPC, forcing that the human player does not influence differently the agents in each test.

C. *Achieved Results*

We performed 100 tests with each type of agent, as shown in Table II. We present in Table II the maximum time (Max. Time), the minimum time (Min. Time) min and Average time spent on each game session. All times are represented in game rounds.

TABLE II. PERFECT AND RANDOM AGENTS TESTS

Agent Type	Max. Time	Min. Time	Average
Perfect	18	10	14
Random	752	594	708

The perfect agent showed a low standard deviation (*i.e.* 4) while the random agent had a high standard deviation (*i.e.* 104). Note that the perfect agent takes less time to win the game without any interference from the human player, while the random agent takes considerably more time in the same scenario. This large time difference occurs primarily by choosing wrong cards unnecessarily by Random agent.

On the other hand we evaluated how the affective NPC reacted according a specific emotion. Other 100 tests were performed for each of the 11 positive emotions that can affect the NPC, as shown in the Table III:

TABLE III. AFFECTIVE AGENT – POSITIVE EMOTIONS

Affective Agent			
High emotion	**Max. Time**	**Min. Time**	**Average**
(+) Gratification	18	10	14
(+) Pride	26	16	22
(+) Happy-for	46	32	40
(+) Joy	82	66	76
(+) Love	88	80	84
(+) Admiration	92	86	90
(+) Gratitude	106	96	104
(+) Hope	112	102	108
(+) Satisfaction	132	116	128
(+) Relief	156	138	140
(+) Gloating	202	166	180

As observed in Table III we can state that some emotions produce a long term effect in the NPC behavior such as “Gloating”, which has the maximum time 202, a minimum of 166 moves and 180 moves on average. In the opposite situation we can find emotions that produce short term effects in the NPC such as “Gratification”, with a maximum of 18 plays, minimum 10 moves and averaged 14 moves. The distribution in the table follows the same order of positive values detailed by Gebhard [4].

As the applied logic does not consider the moves of the human player, but only the moves of the affective NPC, we guarantee that the game was only affected by affective NPC and can better evaluate its performance. Dealing with time values for positive emotions, it seems that depending on the emotion the affective NPC can provide a Perfect NPC behavior or a Random Agent behavior. For example, if we consider the best positive emotion (*i.e.* Gratification) the affective NPC tends to present a behavior very similar to the perfect agent. On the other hand if we consider the worst positive emotion (*i.e.* gloating) it tends to perform like a random agent. From the time values generated we can conclude that the better the emotion, the more likely the agent behavior to maximize your chances of winning. Other 100 tests were performed for each of the 11 negative emotions possible affective agent, as shown in the Table IV:

TABLE IV. AFFECTIVE AGENT – NEGATIVE EMOTIONS

Affective Agent			
High emotion	**Max. Time**	**Min. Time**	**Average**
(-) Hate	226	204	210
(-) Anger	270	552	410
(-) Disappointment	298	268	280
(-) Remorse	318	296	302
(-) Shame	366	322	342
(-) Fear	398	372	382
(-) Reproach	434	404	418
(-) Resentment	488	464	472
(-) Pity	568	508	536
(-) Distress	602	668	630
(-) Fears-Confirmed	698	560	632

The observed values in Table IV allow us to state that some emotions such as "Fears-Confirmed" present a long-term effect in the NPC, with the maximum being 698 moves, a minimum of 560 moves and 632 moves on average. On the opposite we have "Hate", with a maximum of 226 moves; minimum 204 moves played and averaged equals 210. The distribution in the table follows the same order of negative values detailed in the PAD model proposed by Gebhard [4]. As the applied logic does not consider the moves of the human player, but only the affective NPC moves, we have the assurance that the game was only affected by affective NPC behavior and thus we can better evaluate its performance. From the values generated we can conclude that for negative emotions, the worse the value of emotion, nearest agent behavior to minimize the chances of gain. Plotting and sorting the data in Table IV, it is clear a large variability of affective agent, while for the perfect agent and the agent that random variability is non-existent. In Figure 11, the vertical axis in the graph represents the distribution of time spent terminate the game, while the horizontal axis each of the 22 emotions are grouped by time values maximum, medium and minimum.

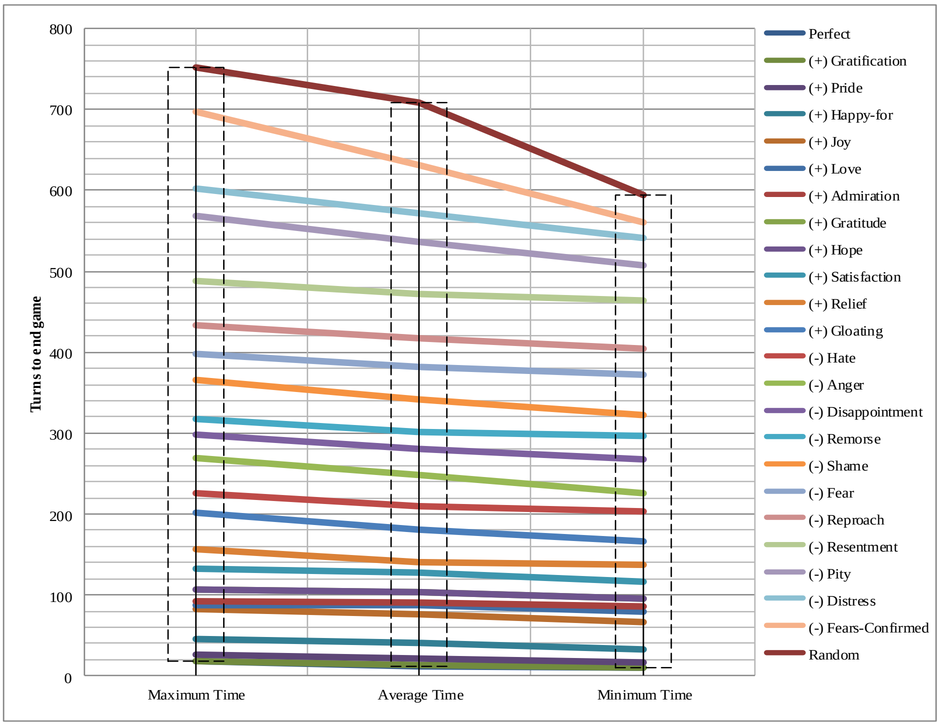

Fig. 11. Variability in the behavior of the affective agent

Using the perfect agent, the average number of rounds to end the game is 12, the maximum number of moves is 18 and the minimum number is 10. Using the random agent, these values are changed completely: in average 708 moves are necessary to end the match, with a maximum number of moves equal to 752 and the minimum equal to 594. It is clear that the range of such values is high, on average equivalent to

5,900% of difference. This way we can note that the affective agent behavior oscillates between the perfect agent and the random agent, according to the emotion applied, thus events with emotion fears-confirmed tends to make the NPC behavior like random. On the other hand, when the NPC is affected by an emotion such as gratification it tends to play near to the perfect agent. In order to understand the growth of the values of affective performance of the agent according to the emotion used, Figure 12 illustrates the results, where the vertical axis present values of time taken to finish the game, and the horizontal axis used emotion agent, lines formed on the graph represent the maximum time (blue), medium time (red) and the minimum time (green).

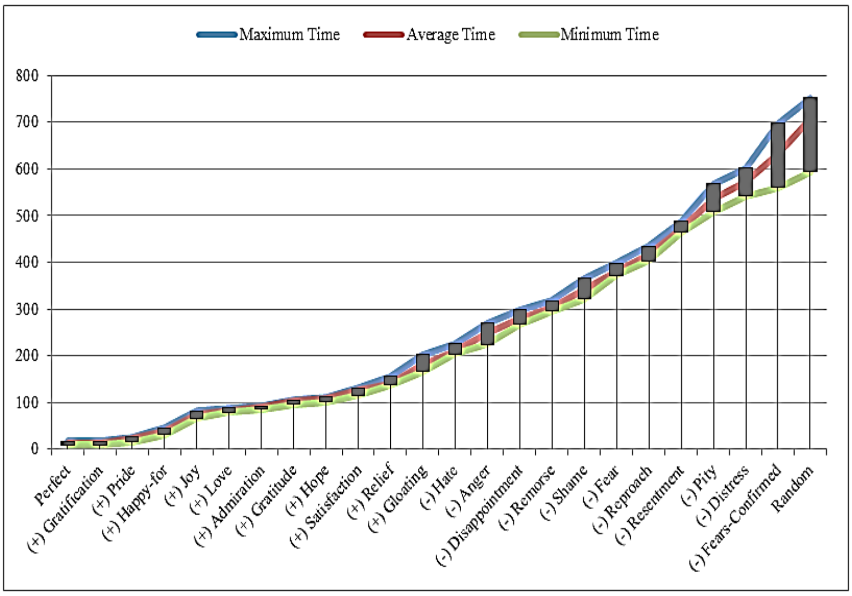

Fig. 12. Gradual increase in the affective agent

From the analysis of the chart and the data base obtained by the tests, we can assume that using the architecture we may have a great variety of different behaviors of the agent. These behaviors vary between a predictable behavior (perfect agent) and a not predictable behavior (random agent).

VII. CONCLUSION

The approach presented in this paper follows the idea that there is a trend to add veracity to the games in order to make them more "real". The approach adopted here goes further, as it not only insert changes of the external environment, but also regards internal changes in the game as consequences of external events. Therefore, introducing change in events according to simulation of emotions is not only attractive in real world simulated games, but it is also helpful in toy games.

We believe that the architecture presented here, based on representing NPC in games as affective agents, may make games more realist (and funny as consequence), from the point of view of the player, since the internal and external events are going to have a direct impact on the performance of NPC. As summarized by Reggie Fils-Aime, the president of Nintendo© "it is not just about graphics. It's about the experience as a whole [12]". From the results obtained with the memory game presented here, it was possible to realize a new level of realism added to the game, which indicates an increasing of the degree of reality within the game. From the scientific point of view, the models presented here are plausible since they are already used in psychology, thereby one of the contributions presented here is an implementation based on them and a test of their real applicability. From the point of view of the architecture presented here, significant results are found that can indicate a new path in the configuration of NPC. This new mechanism with the insertion of personality mood and emotion can generate NPC acting as affective agents, a recent concept in the area of digital games. The use of affective agents in games can open ways of user interaction, since he is going to play a less predictable game.

As feature work we can suggest some points, with the architecture created new steps can be considered For example, new tests should be made with the end user, which looks like fun, not predictability, interaction with the external world to the game, and others should be evaluated and tested seeking create a new version of games. However, to prove whether or not there was an increase of the funny factor, it is necessary to experiment directly with people playing with and without the architecture presented here. Another point of interest is to expand the use of architecture to affective multi-agent and understand the influence of such an environment and among them.

ACKNOWLEDGMENTS

This work was partially funded by FAPERJ and NVIDIA.

REFERENCES

[1] E. Hudlicka, “Affective game engines: motivation and requirements,” in Proceedings of the 4th International Conference on Foundations of Digital Games, 2009, pp. 299–306.

[2] E. Hudlicka, “Affective computing for game design,” in Proceedings of the 4th Intl. North American Conference on Intelligent Games and Simulation (GAMEON-NA), Montreal, Canada, 2008.

[3] S. Kshirsagar, “A multilayer personality model,” in Proceedings of the 2nd international symposium on Smart graphics, 2002, pp. 107–115.

[4] P. Gebhard, “ALMA: a layered model of affect,” in Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, 2005, pp. 29–36.

[5] Z. Kasap, M. Ben Moussa, P. Chaudhuri, and N. Magnenat-Thalmann, “Making them remember—emotional virtual characters with memory,” Computer Graphics and Applications, IEEE, vol. 29, no. 2, pp. 20–29, 2009.

[6] A. Mehrabian, “Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament,” Current Psychology, vol. 14, no. 4, pp. 261–292, Dec. 1996.

[7] A. Ortony, The cognitive structure of emotions. Cambridge university press, 1990.

[8] H. B. Cattell and H. E. Cattell, The 16 PF Cattell Comprehensive Personality Interpretation Manual. Institute for Personality and Ability Testing, 1997.

[9] A. Fioravanti, “Propriedades Psicométricas do Inventário de Ansiedade Traço-Estado (IDATE). 2006. 66 f,” Dissertação (Mestrado)-Departamento de Psicologia, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

[10] R. Sales, E. Clua, D. de Oliveira and A. Paes. "An architecture coupled with a provenance model for affective simulations in games based on real world" (Eds.): SBGAMES 2013 (in press).

[11] R. W. Picard, Affective computing. MIT press, 2000.

[12] “Presidente da Nintendo volta a afirmar que diversão é mais importante que gráficos,” Baixaki Jogos. [Online]. Available: http://www.baixakijogos.com.br/noticias/35899-presidente-da-nintendo-volta-a-afirmar-que-diversao-e-mais-importante-que-graficos. [Accessed: 02-Sep-2013].