iCat, the Chess Tutor
An Affective Game Buddy Based on Anticipatory Mechanisms

Iolanda Margarete dos Santos Carvalho Leite

Resumo da Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Orientadora: Doutora Ana Maria Severino de Almeida e Paiva

Setembro 2007
ABSTRACT
As the co-existence between humans and machines increases, many researchers are attempting to create believable socially interactive characters, robotic or virtual, that can collaborate with humans in a variety of domains and tasks. One of the application domains where these characters have been employed is education, where a large number of pedagogical characters cooperate with users behaving as assistants and companions, giving clues or immediate feedback to the learning experience. In this paper, we study the role of emotions and expressive behaviour in socially interactive characters employed in games. More specifically, on how we can use the character’s emotional behaviour to help users to better understand the game and thus the learning situation. We developed an emotion model for these characters, which is mainly influenced by the current state of the game and is based on the emotivector anticipatory mechanism.

We have implemented the developed emotion model in a social robot named iCat, using chess as the game scenario. A preliminary evaluation was performed, where the robot behaved as a game buddy, playing chess with the user and reacting emotionally to the moves played in the chessboard. The results of that evaluation suggested that the emotional behaviour embedded in the character indeed helped the users to have a better perception of the chess game, which eventually leads them to improve their chess skills.

Keywords
Interactive Characters, Social Robots, Emotion, Anticipation Educational Games.

1. INTRODUCTION
With the rapid development of tutoring systems, educational software simply based on question-answer was not enough. Sample characters behaving as assistants, giving clues and stimulating the user to execute certain actions started to appear in the learning environments. The learning environments many times comprise educational games and the pedagogical character’s purpose is to help learners to perform better in the game [14]. The introduction of embodied characters in these environments positively affects the way that students perceive the experience, especially if those characters convey emotional responses to the learning situation, as they lead to more natural and meaningful interactions [3].

Emotional expression has been considered one of the primary means to achieve believability in synthetic characters, as it helps to know that characters really care about what happens in the world [2]. This assumes special importance when socially interactive characters are immersed in tutoring situations, where the interaction must be motivating and appealing under the risk of losing the user engagement [21]. Even though, many times the emotional responses have been left for secondary role in these systems, being just confined to a couple of predefined animations played in the end of the interaction, which vary depending on the user success or failure in the task. Moreover, in most part of the systems, the character’s emotive responses strongly depend on the learning situation itself and cannot be detached from it, which means that they could not be employed in other games or learning scenarios.

The work presented in this paper addresses the challenge of creating the emotional state and consequent behaviour of a character immersed in an educational game scenario. The main question is the following: how can the character’s emotional behaviour help users to better understand a learning situation? Given this, and since we are going to focus our research in educational games, we start by the hypothesis that if the character acts as a tutor/game companion to the user who is interacting with it, and its emotional behaviour reflects what is happening in the game, users will be able to better perceive the game state and their performance will increase.

This paper is organized as follows. In the next section, we provide a brief literature review of pedagogical agents, virtual or robotic, with emotions. Then, we present the overall agent architecture that we developed to endow pedagogical characters with emotions. Finally, we describe a preliminary experiment carried on to evaluate the effect of the developed model in the user’s perception of the game.

2. RELATED WORK
Emotion plays a key role on routine tasks such as learning, communication, and even rational decision-making [7]. Even so, the incorporation of emotion in machines has been left aside for a long time. Only in the past decade did significant studies on this field started to appear, motivated by Picard’s book “Affective Computing” [20]. The term affective computing refers not only to the ability of a device to detect emotional information from the human who is interacting with it, but also to the ability of expressing an emotional state. Emotional state can be expressed in multiple ways. If the device is a synthetic character with human-like features, the most common ways to express emotions are through speech, facial expression and/or body gesture.

Elliot, Rickel and Lester argue that pedagogical agents would be more effective teachers if they care about the students and convey enthusiasm for the subject matter, in order to foster similar enthusiasm in the student [10]. They believe that this can be achieved by incorporating emotional traits in the characters. Steve and Herman the Bug are two characters that were integrated in the Affective Reasoning Platform developed at DePaul University in Chicago for including personality and emotional responsiveness in virtual tutoring agents [9].

Characters with robotic/physical embodiments that follow the human social rules of interaction are usually named social robots. Endowing these robots with emotions can be very useful for a variety of reasons: (1) to facilitate human-robot interaction; (2) to provide feedback to the user, such as indicating the robot’s internal state, goals and intentions; (3) to act as a control mechanism, driving behavior and reflecting how the robot is affected by different factors over time [11]. A study using the robot Kismet [5] demonstrates that endowing robots with social skills and emotions has benefits far beyond the interface value for the person who interacts with him. Although many researchers bring up the idea of the social robot as a learning companion that interacts with the human learner as a supportive peer or as a tutor [4] [16], there are few practical applications on this domain. This might happen because the whole social robots area is still in an early stage of development. Nevertheless, some efforts are being done, such as the foundation
of the Institute for Personal Robots in Education [13] and the work of Kanda et al. from Osaka University with Robovie [15]. To date, the existing studies involving affective pedagogical characters do not present concrete results on how such emotional traits were relevant (or not) to an increased perception of the learning subject. Still, researchers argue that since motivation is a critical ingredient in learning, and emotions play an important role in motivation, the incorporation of emotions in pedagogical characters should improve student’s learning experience. However, we believe that emotions can bring much more than just motivation to these scenarios. In fact, this is the aim of our work.

3. OVERALL ARCHITECTURE
To investigate about the role of emotions in learning scenarios we developed a scenario composed by a social robot from Philips Research, the iCat [6], which acts as the opponent of a user (preferably a child) in a chess game. The goal is to enable children to play against the character, in a scenario in which the character’s emotive behaviour is influenced by the game state. Children may then interpret the character’s affective behaviour and by doing so acquire additional information to better understand the game.

![Figure 1. Game Flow](image)

In this section, we present the autonomous agent’s architecture behind the character with emotional behaviour that can act as a tutor/learning companion. The architecture was primarily conceived for turn-based games, where the flow is portioned into well defined turns or rounds. In this type of games (e.g. chess), each player has a period of analysis and thinking before committing to an action. An encompassing cycle of the game flow of this scenario is depicted in Figure 1. In the first stage of the interaction, the character introduces itself and the game. Then, it invites the user to play. The user can take the time she wants to think before actually playing her turn. After the user’s turn is over, it is the character’s turn. Here, there is a preliminary phase to appraise the changes in the game, followed by the adjustment of the emotional state as a consequence of those changes. Finally, the character expresses its emotional state and plays its turn. Note that the user may not always be the first one to play. In that case, the steps corresponding to the character’s turn take place before the user’s turn. The cyclic game flow continues until one of the players wins the game.

The agent’s overall architecture is depicted in Figure 2. The architecture intends to assemble the different parts of reasoning in the mind of the agent. Thereby, the architecture model is separated in three different modules: game, emotion and animation. Each module will be better described in the next subsections.

![Figure 2. Agent’s Overall Architecture](image)

Although, in some cases, the emotional state can influence the decision process (as shows the dashed arrow in the figure), we are going to separate those two stages to simplify the model. A very important issue taken into account was the independence of the emotional component in relation to the learning scenario and the character’s embodiment, to ensure enough flexibility to employ this emotional model in other learning scenarios and even different character embodiment.

3.1 Game Module
The game module represents the interactive game played by both the user and the agent. It contains the core dynamics of the game, such as the rules and the algorithms that evaluate the game state. Broadly speaking, this module perceives the game events and selects the actions of the character in the game.

This module also works as the main input for the emotion module, as the character’s affective state is determined by the evolution of the game score. To properly communicate with the emotion module, the game module needs to fulfill a set of requirements. For instance, after each user’s turn, information about the game must be sent to the emotion model. To this information, we will call board value. The board value estimates, after the game evaluation, the value of the character’s position. The game evaluation can be calculated only by looking at the current state of the board or, using a search algorithm, by estimating some moves ahead. In both cases, if the agent is in a better situation than its opponent, the resultant values should be positive, otherwise they should be negative. Moreover, higher values may indicate a greater advantage than values close to zero, and the same for negative values.

In our implementation (the chess game) the game module is composed by two main parts: a user interface and a chess engine [19]. The interface enables users to visualize the chess board and interact with it, for instance, to play their moves. In this case, the user interface was an electronic chessboard provided from DGT Projects [8], which is connected to the computer through a USB interface, and the iCat robot, responsible for communicating with the user to indicate the moves chosen by the chess module. The chess engine is the “thinking” part of the chess application. It
comprises an internal board representation and a set of search techniques and evaluation methods that evaluate a board position and return a move (which the engine considers as the “best one”, given the evaluation) for the character to play. The chess engine implementation is based on Tom Kerrigan’s work [25].

3.2 Emotion Module

The emotion module comprises the representation of the character’s affective state. It receives the board values from the game module and after processing those values the affective state is updated. This update takes effect in the two components of the agent’s affective state: instant reactions and mood.

Our model is inspired by Scherer’s work [23], which separates the affective states in five categories: emotion, mood, interpersonal stances, attitudes, and personality traits. Our agent’s affective state incorporates the first two categories, emotion (that we called instant reactions) and mood. We decided not to include the remaining three categories because, at this time, we did not have enough information from the outside world of the agent that allows us to properly represent those states.

3.2.1 Instant Reactions

Instant reactions are the immediate emotions experienced after the user’s turn in the game. According to Scherer [23], emotions are relatively brief episodes of response to the evaluation of an external or internal event as being of major significance. Although they have a short duration, they are quite explicit.

Instant reactions can be associated with previous expectations, particularly in turn-based games, where we unintentionally and inevitably build an idea of our opponent’s performance. This idea becomes clearer, for instance, when we play with the same opponent more than once. Likewise, the more we think we “know” our opponent, the more we get surprised with her failure if we consider her a great player. In other words, we tend to anticipate our opponent’s performance during the game.

To endow our agent with this kind of behavior, we used an anticipatory system named emotivector. An anticipatory system is a system containing a predictive model of itself and/or of its environment that allows it to change state at an instant in accord with the model’s predictions pertaining to a later instant [22]. The emotivector is an anticipatory system that generates an affective signal resulting from the mismatch between the expected and the sensed values of the sensor to which it is coupled to [17]. In our architecture, the emotivector’s sensed values correspond to the board values, whereas the expected values are expectations of those board values, computed using a prediction algorithm. When a new board value is received in the emotion module, the emotivector system catches this value and performs the following set of actions: (1) using the history of board values, the next expected value is computed; (2) by confronting the expectation computed in the previous step with the actual sensed value (i.e., the new board value), and using a model inspired in the psychology of emotion and attention, the emotivector computes a sensation for the percept.

The nine possible sensations resultant from the emotivector model are presented in Figure 3. For each group of expectations (expected reward, negligible expectation and expected punishment) three different sensations can be elicited, depending if the sensed value is higher, lower or within a threshold value. This threshold is also computed with a prediction algorithm, using the history of mismatches between the expected and the sensed values. Predicting the threshold value has the advantage of not requiring fine-tuning, which is fundamental for our model, since the signals picked up by the emotivector can vary depending on the game that is being used. Thus, the only adjustment that needs to be done when implementing this model is on defining the negligible variation for the nine sensation model.

![Figure 3. Emotivector: nine sensation model](Image)

For instance, after three moves in the chess game, if the iCat has already captured an opponent’s piece, it might be expecting a reward for the user’s next move. So, if the user actually plays a bad move (e.g., by putting her queen in a very dangerous position), the elicited sensation will be a “stronger reward”, which means “this sensation is better than I was expecting”. On the other hand, if the iCat is expecting a reward, and the opponent captures an iCat’s piece, the elicited sensation will be a “weaker reward”, which means “this sensation was worse than I was expecting”.

Different sensations can be elicited and therefore different instant reactions are sparked. The selected sensations will then pass to the animation module, where they are directly mapped to an affective expression that reflects a certain emotion.

The Prediction Algorithm

As mentioned earlier, to calculate the expected board value and the threshold for the emotivector system, a prediction algorithm must be applied to the history of those groups of values. We have used the moving averages algorithm [18]. Moving averages’ underlying purpose is to smooth a data series and make it easier to spot trends. This algorithm is commonly used in the stock market.

The two most popular types of moving averages are the simple moving averages and the exponential moving averages. We have applied the exponential moving averages, also called weighted moving averages, which reduces the lag by applying more weight to recent values relative to older values. The formula is the following:

\[
\hat{S}(n) = (1 - \alpha) \cdot \hat{S}(n - 1) + S(n)
\]

\[
\alpha = \frac{2}{1 + N}
\]

For a certain move n, \(\hat{S}(n)\) represents the expected value, \(\hat{S}(n - 1)\) is the previous expected value and \(S(n)\) is the current sensed value. \(\alpha\) is a smoothing factor and \(N\) is usually the specified number of periods. When using the formula to calculate the first \(\hat{S}\), there is no available value to use as the \(\hat{S}(n - 1)\). This small problem can be solved by starting the calculation with a simple moving average and continue on with the above formula from there. A simple moving average is formed by computing an arithmetic mean over a specified number of periods.
3.2.2 Mood
Mood is a relatively lasting affective state. It is less specific, often less intense and thus less likely to be triggered by a particular stimulus or event. Moods generally have either a positive or negative valence effect and are longer lasting [24]. Valence refers to the emotional value associated with a stimulus. Mood works like a background emotional state, when other emotions (in our case, the instant reactions) are not occurring.

Based on the above definition, we will represent mood as a valence variable V that ranges between [-100; 100]. The magnitude of V represents its intensity. Positive values are associated to good things (happiness), whereas negative are related to bad things (unhappiness). After its calculation, valence will be handled by the animation module, which will turn the value into the corresponding embodiment expression. The board values are also the main stimuli for the valence variable. However, to do so, some pre-processing is required. Firstly, because the limits of the board values may not lie between the [-100;100] interval. Secondly, even if the previous statement is true, this mapping only leads to linear correspondences, which cannot be desirable in some cases. Suppose that the values sent from the game module correspond to an evaluation function that ranges from -100 to 100, where -100 means that the agent is going to lose in the next turn and +100 means the agent is certainly going to win. In this type of functions, these boundary values only come up in the endgame. Thus, most part of the signals will belong to [-30; 30], especially in the beginning of the game. Linearly mapping these values would lead us to low intensity valence at the most part of the game. To overcome this kind of situations, we introduce an intermediate filter function F(x), which will filter the values received directly from the game module before they reach the valence variable. In that function, x corresponds to the values received from the game module. In the chess game scenario we have employed the below filter function, which has a graphical representation plotted in Figure 4.

\[
F(x) = \begin{cases} 
\log(x + 1) * 25, & x > 0 \\
0, & x = 0 \\
-\log(-x + 1) * 25, & x < 0 
\end{cases}
\]

**Figure 4. Mood’s Filter Function**

Summarizing the mood computation steps, when a new board value is received by the emotion module, this value gets into a filter function. After that, a valence value (between -100 and 100) is sent to the animation module. Although mood is defined as a lasting affective state, it does not last forever. Over the time, and in the absence of new stimuli (i.e., new board value signals), valence will decay and towards zero again. A decay rate, which indicates how fast the valence decays over time, as well as the number of seconds before the decay starts since the last stimuli, must be defined.

3.3 Animation Module
The main role of the animation module is to display the internal affective state of the agent to the user, through the manipulation of the character’s body. Thus, this module is associated to an embodiment composed by a set of variables that represent different body parts (e.g. left_eyebrow). Variables have a name and a range value. The current value of each variable will then be reflected in the embodiment of the character. There are two different ways to manipulate these variables: direct manipulation or predefined animations. Direct manipulation sets the value of a single variable in real time, whereas predefined animations are scripts containing a temporal sequence of variables and their values (i.e. several direct manipulations), which result into movements of those body parts. The predefined animations can be used to represent emotional behaviours such as happy, sad or surprise, but also broader animations like blinking, looking down or looking up. This module manages three main groups of data. Each one fits into one of the two categories defined above:

**Emotions.** The animation module is responsible for the mapping between the three sensations resultant from the emotivector system and the corresponding predefined animations in the character’s embodiment.

**Mood.** While emotion animations belong to the predefined type, mood follows the approach of direct manipulation. Two different sets of variable parameterizations must be defined, a positive and a negative, each one corresponding to a limit of the valence space (+100 and -100). The embodiment position is determined as an interpolation from one of those two parameterizations. If valence is positive the values are computed using the positive sets of variable parameterizations, whereas the negative valence values are
computed through the negative parameterization. For example, suppose that the positive parameterization contains a definition for a variable named eyebrow with the value 80. If the valence equals 50, the value of this variable should be half of the parameterized value. With this mechanism, we can display the same expression, but with different intensities. The animation module is also responsible for smoothing the values of the body variables, by extending those changes for several cycles ahead.

**Other animations.** The affective state itself is not enough to produce believable behaviour. The emotional animations are of short duration (and only played after the user plays a move) and the mood only changes gradually the values of some character’s body parts. Thus, there are some periods where the character seems to “freeze”, destroying the user’s suspension of disbelief. For this reason, the animation module also performs idle animations, such as blinking or looking to sides randomly, to increase the character’s overall believability. Like instant reaction animations, idle animations are also predefined. This category also incorporates the animations sent by the game module, which correspond to the actions taken by the agent during the game. The game actions are restrained to speech acts, which in the chess game are used by the iCat to ask the user to play its moves in the chessboard.

The inputs for the animation module are the sensations triggered by the emotivector system and the valence variable computed by the mood component, both from the emotion module. The following set of actions is performed continuously by this module: (1) if a new sensation is thrown by the emotivector system, the corresponding emotion is played through a predefined animation; (2) the body variables associated with the mood parameterizations are updated, according to the current valence value; (3) if the embodiment is not playing any predefined animation from the step 1, a random “idle animation” is performed.

After the three steps, some variables may be used concurrently, which may lead to conflicts. To solve this problem, a priority was established to access the variables. In the presence of overlapping values, the most priority values are the ones set up by instant reactions, followed by the mood and finally the other animations.

3.3.1 **Animating the iCat**

The animation module is the part of the architecture responsible for communicating with the character’s embodiment to convey the agent’s affective state to the user. In our implementation scenario, this module communicates with the iCat control software, the OPPR Platform, which controls the robot’s facial expressions [6]. The OPPR platform provides an animation engine which makes it possible to combine multiple robot animations at runtime execution. Such combination is possible due to the existence of ten animation channels with different priorities. In the iCat’s terminology, the term animation does not only refer to the movement of mechanical parts, but also to changes in light, sound and speech [6]. Besides the animations, this engine also renders behaviours, which are “dynamic” animations written in Lua Programming Language [12] that also change the values of the iCat’s body parts. The iCat software also offers a lip synchronization mechanism that works as a regular animation, played in a certain channel. This particular animation plays an important role in the character’s believability because it ensures the synchronization between mouth movements and acoustic speech.

We used the robot predefined animations from the animation library of the OPPR software for the emotions and the idle animations, and for the mood we used a behaviour that allows us to directly manipulate the value of some variables in the robot’s embodiment. The OPPR library animations were previously submitted to tests which verify that users perceive those emotional expressions on iCat’s embodiment [1]. We have made a correspondence between these animations and the nine emotivector sensations, as depicted in Table 1. The emotional animations were not enough to map all the sensations. Thus, we adapted other animations (such as apologize or confirm) to some sensations. The animations marked with “*” were slightly modified in order to recreate an emotion. The choices for the sensation-animation mapping were based on the meaning of the sensations. For example, the “stronger reward” sensation means that we experienced a reward much better than we were expecting and therefore the correspondent emotion is “excitement”.

<table>
<thead>
<tr>
<th>Sensation</th>
<th>Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td>stronger R</td>
<td>excited</td>
</tr>
<tr>
<td>expected R</td>
<td>confirm*</td>
</tr>
<tr>
<td>weaker R</td>
<td>happy</td>
</tr>
<tr>
<td>unexpected R</td>
<td>arrogant</td>
</tr>
<tr>
<td>negligible</td>
<td>think*</td>
</tr>
<tr>
<td>unexpected P</td>
<td>shocked</td>
</tr>
<tr>
<td>weaker P</td>
<td>apologize*</td>
</tr>
<tr>
<td>expected P</td>
<td>angry</td>
</tr>
<tr>
<td>stronger P</td>
<td>scared</td>
</tr>
</tbody>
</table>

4. **EVALUATION**

In order to show that the developed emotion model has some impact in the user, we performed a preliminary experiment with the scenario containing the iCat as an opponent of a human player in a chess match. More precisely, we aimed at providing the answer to the following question: *what is the effect of the iCat’s emotional behaviour on the user’s perception of the game?*

The hypothesis is that the users will be able to understand the iCat’s emotional behaviour, and by that they will have a larger perception about what is happening on the game, and thus be able to improve their chess skills.

4.1 **Methodology**

4.1.1 **Measurements**

With the presented experiment we attempted to measure the user perception of the game. The criteria that we adopted for measuring the success of the user perception of the game was to compare what the user “thinks” about the game at a certain moment, with the value obtained from the chess engine’s evaluation function. As such, at a certain board position, if these two variables match with each other (e.g., if the user thinks that iCat is losing and the chess evaluation function also indicates that iCat is in disadvantage), we say that the user could successfully perceive the game state.
4.1.2 Participants
A total of 9 participants, 7 males and 2 females, between 7 and 31 years old took part in the experiment. There were two criteria used when selecting the participants. First, none of the participants had prior experience interacting with the iCat. The second criterion was that the players must know all the rules of the game and at least have some prior experience on playing chess. Four of the nine participants (the youngest) were members of a chess club and so they practiced chess regularly.

4.1.3 Setting
We divided the experiment in two different sessions. The first session was conducted in the chess club facilities with the children and the second session was at the IST – Technical University of Lisbon with the remaining subjects.

The experiment was conducted with three different control conditions regarding to the iCat’s emotional behaviour:

1. The behaviour in agreement with the implemented emotional model;
2. “Incoherent” random emotional behaviour. In this case, the instant reactions to the user’s move are randomly chosen between eight possible animations. This means that the only animation that cannot be chosen is the one that would be selected in the “coherent” emotional behaviour. The valence value computed after each user’s move is also a random value;
3. Without expressing any emotional state, that is, a neutral/idle behaviour.

The experiment was composed by three chess problems, with different levels of difficulty (easy, medium and hard). The three exercises were suggested by the chess instructor from the chess club where some of the tests were performed. Participants interacted with the three versions of the iCat, each one in one of the different exercises. The first exercise they were asked to solve was the easy one, followed by the medium and finally the hardest. The control conditions of the iCat’s behaviour were incorporated in a balanced order. This means that one of the participants played the easy problem with the condition number one, the medium with the second condition and the hard with the third. Then, the next participant played the easy problem with the second condition, the medium with the third and the hard with the first, and so on. The non-affective animations, i.e., the ones that iCat uses to communicate its moves to the opponent, as well as the idle animations were exhibited in the three interactions, apart from the control condition.

4.1.4 Procedure
The participants sat at a table in front of both the chessboard and the iCat. On the table, there was also a laptop, connected both to the chessboard and to the iCat. The experimenter was seated next to the participant. Each participant completed the evaluation session in approximately 45 minutes.

First, the experimenter welcomed the participant and explained how to use the chessboard. Since the board has sensors in all the squares, when moving the pieces, these should be lifted up instead of dragged. Next, the experimenter explained to the user that, due to the iCat’s embodiment limitations, she must play the robot’s move, when asked. The interaction begins with iCat performing a “waking up” animation, and after that it invites the user to play. When an exercise terminates, i.e., when one of the players (iCat or the user) checkmates the other or the game draws, the iCat plays the corresponding animation to that event and then falls asleep. After that, the board is set up with the new exercise and the iCat wakes up again. Figure 6 contains a storyboard of the interaction between a user and the iCat during the experiment.

As mentioned above, these answers will then be compared with the board evaluation from the agent’s game module at the time the questions were made. Since that evaluation function returns integer values, we needed to map them into “winning”, “loosing” or “neither winning or loosing” so they could be compared with the user’s answers. The evaluation function returns positive values when the agent is “winning” and negative values when it is “losing”. To map the “neither winning or losing” option, we considered the values where the difference in the score is not significant between the two players (typically the difference of one pawn in the material).

Finally, the users filled a paper and pencil questionnaire containing an open-answer question about the main differences in the iCat’s behaviour among the three exercises, if they did notice such differences.

4.2 Results
4.2.1 User perception of the game
Since we had conducted the tests with nine different subjects and asked them the “perception of the game” questions three times during each exercise, we have a sample of 27 values for each one of the three control conditions.

Figure 6. User playing with the iCat
Table 2. Success statistics of the perception of the game variable

<table>
<thead>
<tr>
<th>Independent Variables</th>
<th>Chess Experience</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Beginner</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Neutral</td>
<td>20</td>
<td>13</td>
</tr>
<tr>
<td>Random emotive vector</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>N</td>
<td>23</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 2 shows the number of success cases organized by iCat’s emotional behaviour and subject groups. The success measure varied among the three different control conditions. Although the sample is not very large, the results are better when the iCat comprised the affective behaviour described (emotivector based). These results suggest that such behaviour helps the users to better understand the game. We separated the subjects in groups based on their chess experience (5 beginners and 4 intermediates) and regarding to their age (4 under 14 years old and 5 above 14). Taking a deeper analysis between the groups, we did not find any abrupt divergence with the success distributions, which indicate that this success is independent from the subject group.

We have also performed a Spearman Correlation test with a Twotailed test of significance for the samples of each one of the three control conditions. We had three different variables to correlate: the user’s perception of the game based on the iCat’s expression, the user’s perception of the game based on her overall analysis and the “actual” game state, obtained from the chess evaluation function.

The most relevant aspect is that the correlation between the “user’s own analysis of the game” and the “actual game state” variables is higher when the control condition comprises our emotional system. The value of the correlation is 0.930 (p < 0.001). With the random emotional behaviour samples, the correlation decreases to 0.485 (p = 0.010) and in the games that iCat did not express any affective state the value is 0.680 (p < 0.001). Since the only aspect that changed among the three sets of values was the iCat’s behaviour, these results suggest that the user’s perception of the game increases when the iCat’s emotional behaviour is in agreement with the actual state of the game, that is, when it is animated with the previously described emotional system.

Regarding to the correlation between the perception of the game based on the iCat’s expression and the user’s own analysis of the game, we found correlations between these two variables in two of the three control conditions. The correlation tests using the values from our emotion system, such correlation is really strong (0.958 for p < 0.001), whereas in the neutral emotional behaviour the value decreases to 0.580 (p = 0.002). Despite the decrease in the value, the variables still remained correlated with the neutral behaviour, which was quite unexpected. One possible explanation for such correlation is that users tend to interpret the iCat’s neutral behaviour taking into account their opinion in the game. Thus, we cannot guarantee a total independence between the variables “user’s perception based on iCat” and “user’s perception based on own analysis”. The exception in the correlation between these two variables took place in the random control condition values, where the variables are negatively correlated (-0.116), although with a poor significance (p = 0.564).

Finally, concerning to the relation between the perception based on iCat’s expression and the real value of the game, it is important to refer that there is a strong correlation between these variables when applying the test to the values retrieved when the iCat’s behaviour is in agreement with the described emotional system (0.980 for p < 0.0001). This result contributes to our belief that users were able to correctly interpret the iCat’s emotional behaviour.

4.2.2 Differences between the iCat’s behaviour

In the youngest group, the one that comprises subjects with less than 14 years old, only one of them noticed differences in the iCat’s behaviour. In the other group, including subjects older than 14 years old, the differences were noticed by all the members. However, some of the members that belong to the “beginners group” regarding to the chess experience, only distinguished two different types of behaviour in the iCat: a “neutral” and “active” behaviour. Therefore, only in the more experienced chess players group from the all the participants correctly distinguished the three different behaviours.

4.2.3 Overall Analysis

During the experiment, some qualitative data was collected through observation. For instance, we observed that participants were quite polite when interacting with the character. For example, after they play the iCat’s move, when iCat said “thank you”, they usually replied with a “you’re welcome” sentence. Such responses strengthen the feelings that users saw the iCat as a believable character.

In general, children paid more attention to the iCat than adults. Some of the older participants were so concentrated in the game that they almost forgot to notice the iCat. On the other hand, children expressed much more anxiety and enthusiasm about the iCat, not only after the instant emotional reactions, but also during the entire game. Despite it was the adults group that could better distinguish different iCat behaviours, it was the children’s group who demonstrated more enjoyment while playing.

5. DISCUSSION AND CONCLUSION

The work presented in this paper intends to be a step towards the development of characters with emotional behaviour that can act as companions to facilitate a child’s own efforts at learning. Such characters shall not be compared to the traditional question-answer based intelligent tutoring systems, neither with conventional human tutors. Instead, they must be perceived as more experienced companions that can help users to accomplish a task or improve a particular skill. One of the reasons why the emotional expressiveness in pedagogical characters is so important, aside from the increased believability is that people typically assume that emotions are spontaneous and therefore they are an honest way of communication between them and the agent.

In the preliminary experiment that we conducted, the subjects succeeded more times in the perception of the game measure when the iCat’s emotional behaviour was in agreement with our emotional system. Furthermore, the correlation between the user’s analysis of the game and the actual state of the game variables was substantially higher in this case. From the qualitative data, we could retrieve information about how users perceived the iCat’s behaviour. While most of the subjects were able to identify the idle affective behaviour, only a part of them could correctly distinguish the three distinct behaviours. The problems on distinguishing the emotional random behaviour from the
“coherent” one can be justified by the fact that younger participants are used to play with players that commit mistakes. They did not find iCat’s behaviour incoherent, but instead they thought that the robot was performing a bad evaluation from the game state. These results suggest that, despite the disconnection between the character’s behaviour and the actual game state, it was coherent in some way, and therefore it was acceptable for children.

Although preliminary, the results of this experiment indulge us to believe that we are in the right direction to build the emotive behaviour of an artificial game-playing agent that can actually help users to better understand the game, and by that help them to increase their performance in that game.

6. ACKNOWLEDGMENTS
Our thanks to DGT Projects for providing us with their DGT electronic chessboard.

7. REFERENCES