Emotional Behavior Trees

Anja Johansson and Pierangelo Dell’Acqua

Abstract—Often the AI techniques for decision making used in commercial games are predictable and unadaptive. Arguably, this causes a lack of realism for the players. We believe that emotions are a vital part in the creation of interesting and believable non-player characters for games. In this paper, we present an extension to behavior trees that incorporates emotions into the decision making. Specifically, we introduce a new type of priority selector whose priorities are dynamically evaluated to allow for emotional influence. This selector takes into account time-discounting, risk perception, and planning as relevant factors of the decision making that emotions greatly influence. The objective of our work is to provide game developers with a technique to model gaming scenarios using emotional behavior trees.

I. INTRODUCTION

In the past years behavior trees have become an increasingly popular action selection mechanism for non-player characters (NPCs) in commercial games. Prior to the use of behavior trees, finite state machines [31] were frequently used. Finite state machines have problems scaling up, however, as the transitions become increasingly difficult to manage when the size grows. Attempts were made to resolve this problem by introducing hierarchical finite state machines [9]. These allow behaviors to be grouped together and to share transitions. Recently, behavior trees became popular because they have excellent properties, they scale up well and they are intuitive and memory-efficient [3]. Games known to use behavior trees include Halo 2 [12], Halo 3 [13], Spore [10], and GTA[6]. The game-engine/rendering-engine CryENGINE by Crytek also uses behavior trees for the AI [33].

Emotions are vital for human decision making. Not only do emotions act as a help to choose quickly between numerous alternatives, but they are in fact a vital and integrated part of the decision making itself [7], [8]. It is not straightforward, however, to couple behavior trees with emotions to mimic human emotional decision making. If we wish to have a natural and interesting behavior, it is important that the characters behave in an emotional way. It could be claimed that it is possible to incorporate emotions into behavior trees by merely using emotions in the conditions. However, doing so may create large cumbersome behavior trees that are difficult to manage. For each behavior, a specific set of conditions would have to be placed on emotional states. These conditions would most likely take the form of checking the emotional values against a fixed threshold, which would disable a subtle emotional effect on decision making. Using this approach would most likely lead to a large behavior tree with numerous nested conditions, making it difficult to construct and manage.

In this paper we present an extension to behavior trees that incorporates emotions. However, we do not wish to lose one of the benefits of the behavior tree, namely the possibility to control the character’s behavior. We want to let the behavior designer choose which behaviors should be under emotional influence. We choose three important factors within decision making that are greatly influenced by emotions: time-discounting, risk perception, and planning. While there are other factors also involved in decision making, these three factors fit well within the scope of the decision making for NPCs in computer games. Using these three factors, we construct an emotional selector (based on a priority selector) that uses emotions to change the priorities of the child nodes of the selector at runtime. A fighting NPC scenario is used to display the contributions of the model.

II. BEHAVIOR TREES

Despite the fact that there is no formal definition, behavior trees have been comprehensively described by Champandard [3], [4], [5], [6] and Knafla [16]. Damian Isla has also discussed the implementation of behavior trees in commercial games [12], [13]. In this section, we introduce a definition of behavior trees based on the description of Champandard and Knafla.

A behavior tree (BT) is a directed acyclic graph consisting of different types of nodes. Most of the time the behavior tree is tree-shaped, hence the name. However, unlike a traditional tree, a node in a behavior tree can have multiple parents which allows the reuse of part of the BT. The traversal of a behavior tree starts at the top node. When a node is executed, it returns one of the three states: success, failure or running. The first two are self-explanatory and running signifies that the node has not yet finished executing. A behavior tree consists of the following types of nodes.

A. Leaf nodes

Action: An action represents a behavior that the character can perform. The action returns the state success or failure when it completes its execution depending on the outcome. When an action needs more time to complete, it returns the state running. An action is depicted as a white, rounded rectangle.

Condition: A condition checks an internal or external state. It returns either success or failure. Conditions are similar to actions except that they execute immediately and hence never return running. A condition is represented as a gray, rounded rectangle.
B. Inner nodes

Sequence Selector: A sequence selector is a node that typically has several child nodes that are executed sequentially. As long as a child node completes its execution successfully, the sequence selector continues executing the next child node in the sequence. If every child node returns success, then the sequence selector returns success. Should one of the child nodes return failure, the sequence selector immediately returns failure. If a child node returns running, the sequence selector also returns running. The sequence selector keeps track of which, if any, of its child nodes is currently running. A sequence selector is depicted as a gray square with an arrow across the links to its child nodes.

Priority Selector: A priority selector has a list of child nodes which it tries to execute one at a time, with respect to the specified order, until one of the child nodes returns success. If none of the child nodes executes successfully, the priority selector returns failure. If a child node returns running, the priority selector also returns running. The priority selector keeps track of which of its child nodes, if any, is currently running. A priority selector is represented with a gray circle with a question mark in it.

Parallel Node: A parallel node executes all of its child nodes in parallel. A parallel node can have different ways of determining when to stop executing its child nodes. One may specify the number of child nodes that must execute successfully for the parallel node to succeed, and likewise the number of child nodes that must fail in order for the parallel node to fail. A parallel node is depicted as a gray circle with a P in it.

Decorator: A decorator is a node that acts as a filter that places certain constraints on the execution of its single child node without affecting the child node itself. For instance, a decorator can prevent a child node from executing more often than every five second. Decorators are represented as diamonds with descriptive text inside.

III. EMOTIONS AND DECISION MAKING

The term affective state refers to emotions, feelings or moods. Their distinction is rather vague and not well-defined; yet generally one may define them in the following way. Emotions are short-term, intense and usually related to a specific event. Feelings are usually classified as emotions that a person has become cognitively aware of. Moods are less intense, more long-term, and not related to a single event, at least not on a conscious level. In this paper, we will not make any distinction between them. In this section, we will describe the impact of affective states on decision making, then we will describe three specific factors that emotions affect which are relevant for our work.

A. Emotions and Decision Making

The importance of emotions for decision making has been studied in a variety of different fields. Previously it was believed that emotions and rationality were separate, and that emotions only affected decision making in a negative way by affecting the rational thinking. This has been proven to be incorrect. Many studies suggest that emotions play a vital role in decision making [7], [8], [17], [35], [36].

LeDoux [17] for example has done extensive research on how emotions, especially fear, function and how they affect memory. Many of his studies [18], [19] have focused on fear-conditioning, which is a type of learning where an organism learns to connect a previously neutral context to something negative. Memories are more easily retrieved when a person is in a similar emotional state to the one the person was in when the events were experienced. This in turn will affect the decision making, since different information is available at different emotional states.

Damasio [7], [8] has studied patients which due to injuries to the prefrontal cortex experience impaired emotional capabilities. While these patients do not seem to suffer from any cognitive disabilities or lack of intelligence in any way, they are insufficient at, even incapable of, making everyday choices. Damasio suggests a somatic-marker hypothesis, where the brain uses emotions to mark different options with the help of emotions and other low-level physical states. This enables the person to make choices fast and efficiently even when presented with numerous alternatives.

![Loewenstein and Lerner model of emotions](image)

In the field of psychology, Loewenstein et al. have suggested a general model of how emotions influence decision making [23]. This model is depicted in Figure 1. According to their model, emotions affect the decision making process in two different ways. In the first, emotions affect how people value the outcome of their actions. In general, humans try to maximize positive emotions while minimizing negative emotions. When deciding what action to choose, we try to determine the emotional consequences of our actions. This group of emotions is called expected emotions and is shown in link a. The second group is concerned with emotions that a person experiences at the time he or she is making the decision. These emotions can be a result of the decision making itself (link b and c). When a person contemplates the alternative actions, emotions related to those actions are triggered. These in turn affect the final outcome of the decision making. The second group of emotions can also be totally unrelated to the task at hand (link g). These are emotions that the person experiences due to circumstances that are not related to the current decision making task.
risks are perceived, how far ahead the person will plan, etc. Directly (link \( h \) and \( i \)), current emotions affect the decision making by affecting individual decisions, trying to determine how the person feels about the different choices. Emotions also influence the reasoning around the outcomes of a specific action (link \( h \)) and the emotions that are the consequences of that outcome (link \( i \)).

Studies by Bless and Schwarz [2] have shown that people who are in a good mood tend to change their opinions more often without considering whether the arguments behind their opinion is strong or weak. For people in a bad mood, the opposite is true. Hence, people who are in a good mood may sometimes switch behaviors without a clear reason or behave erratically.

Emotional states affect also how optimistic people are about future events. Mayer et al. [30] have shown that people in a good mood are more inclined to believe that future events will be in their favor.

B. Risk perception

The perceived risk of an action is greatly influenced by emotions [24]. Studies by Lerner and Keltner [20], [21] have shown that happy and angry people are willing to accept greater risks, while fearful people are more pessimistic. Raghunathan and Pham [34] proposes that anxiety is connected to risk-avoidance, while sadness allows for greater risks, but instead gives a focus on high rewards. A study by Maner et al. [28] also shows that people with anxiety are more prone to risk-avoidance behavior.

C. Time discount

Time plays a great role in decision making. For instance, when a person has a choice between two actions, the first one leading to a semi-satisfying immediate payoff and the second leading to a very satisfying payoff three weeks from now, the emotional state of that person greatly influences the decision that person will take. Since humans tend to consider time spent waiting for the reward as a cost, they often choose the quicker, but smaller reward. Judging time to be cost is called time-discounting. People who more often choose quicker rewards are considered impulsive. We know that emotions affect impulsiveness. People who are in an elevated emotional state (such as people who are very angry or frightened) are more impulsive, which means that they pay less attention to delayed payoffs [23]. In 1976, Moore et al. concluded that people who are experiencing happiness choose delayed rewards more often than people who are experiencing sadness [32]. Wittmann and Paulus conclude that people with depression, people who are bored, and people with a high level of anxiety often have larger time-discounting, since time moves subjectively slower to them [40]. It is worth mentioning that humans often greatly underestimate the impact on their choices on future emotions, despite previous experiences telling them otherwise [22].

D. Planning

Schwarz and Clore present theories on “feelings-as-information” [35], [36], [37], [38]. In general, this concerns how emotions are used by humans to make judgments on things they are faced with. It is known that sadness makes us focus on goals that are tied to rewards, whereas anxiety focuses on goals that reduce the risks. Sadness signals a problem in our world and triggers a systematic processing (bottom-up processing, attention to details), whereas happiness enables a simpler, top-down processing. Luce et al. [27] have also shown that humans in a bad mood tend to focus on bottom-up processing. Humans in a bad mood also focus on one attribute at a time, making the decision making process more cumbersome and time-consuming. This will inhibit the amount of planning a human can perform. Studies by Lorist [26], [25] as well as by van der Linden et al. [39] have shown that mental fatigue also has a negative impact in information processing and planning.

IV. Emotional Behavior Trees

To take emotions into consideration, we extend the definition of behavior trees and introduce a new type of selector, called the emotional selector. We call the resulting model the emotional behavior tree (EmoBT). The emotional selector orders its child nodes according to three factors: the planning effort, the risk involved, and the time it takes to perform the actions. The formulas for the EmoBT are constructed using an intuitive approach as psychology findings are not specific enough to draw more precise conclusions. Thus, these formulas can only be empirically validated in test case scenarios.

We assume the current affective state of the NPC is known to the EmoBT. The emotions may be modeled as desired, but must be represented by a value in the interval \([0, 1]\).

A. Planning effort

In contrast to planning processes, a behavior tree specifies all the actions that must be executed under certain circumstances. In order to mimic the effort of a planning process, as if planning would have taken place, we introduce the notion of planning value.

Action: An action has a planning value that is set by the designer (1 by default). Some actions, such as complex pathfinding, involve more planning than others and it is therefore important that the designer can specify the amount he/she thinks is a reasonable estimation.

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

Sequence Selector: A sequence selector \( j \) is designed to perform all its children in the sequence. This can be thought of as plan. The planning value is defined as:

\[
\text{plan}_{j} = \sum_{i=1}^{N} \text{plan}_{i}
\]

where \( N \) is the number of child nodes and \( \text{plan}_{i} \) is the planning value of its child node \( i \).
**Priority Selector:** Since one cannot know which of the child nodes a priority selector \( j \) will perform, the planning value of \( j \) is defined as an average value:

\[
plan_j = \frac{\sum_{i=1}^{N} \text{plan}_i}{N}
\]

where \( N \) is the number of child nodes of \( j \). Ideally, if available, the probability of a successful execution of every child node should be taken into account in this equation.

**Parallel Node:** Since a parallel node performs all of its child nodes, the parallel node must have planned all child nodes before executing. Hence the planning value of the parallel node \( j \) is defined as:

\[
plan_j = \sum_{i=1}^{N} \text{plan}_i
\]

where \( N \) is the number of child nodes of \( j \).

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

### B. Risk Assessment

Risk has to do with how dangerous the character believes a situation is. A risk value is between 0 and 1; 0 being no risk at all, and 1 being extremely dangerous. The risk value measures the probability of risk. EmoBTs cannot reason about the risk of performing an action, but we allow the designer to add a risk value to each leaf node in the tree, and derive the associated risk for the inner nodes.

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

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

**Sequence Selector:** Since a sequence selector performs every child node of the sequence, the risks of every child nodes must be combined. The overall risk value is calculated as:

\[
risk_j = 1 - \prod_{i=1}^{N} (1 - risk_i)
\]

where \( N \) is the number of child nodes of \( j \).

**Priority Selector:** A priority selector \( j \) only executes one of its child nodes. Since we cannot determine in advance which node will be executed, we define the risk value of \( j \) as the average of the risk of every child node \( i \):

\[
risk_j = \frac{\sum_{i=1}^{N} risk_i}{N}
\]

**Parallel Node:** Since all of the child nodes of a parallel node \( j \) are executed, the risk is defined as:

\[
risk_j = 1 - \prod_{i=1}^{N} (1 - risk_i)
\]

where \( N \) is the number of child nodes of \( j \).

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

### C. Time

When making a decision, it is important to take the execution time of actions into consideration, especially under certain emotional conditions. Every node in the EmoBT has a time interval \([L, U]\) (with \( L \leq U \)) representing the lower and upper limit of the time needed to execute that node. The time interval for the execution of leaf nodes must be given, while for inner nodes it is derived.

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

**Condition:** A condition has a time interval of \([0 0]\) because conditions are always instantaneous.

**Sequence Selector:** The execution time interval of a sequence selector \( j \) is the sum of the lower and upper limits of the time intervals of every child node \( i \):

\[
[L_j, U_j] = \left[ \sum_{i=1}^{N} L_i, \sum_{i=1}^{N} U_i \right]
\]

where \( N \) is the number of child nodes of \( j \).

**Priority Selector:** A priority selector \( j \) has the option to choose between its child nodes. Since we cannot know which child will be chosen, we must look at the overall time interval:

\[
[L_j, U_j] = [\min\{L_1, \ldots, L_N\}, \max\{U_1, \ldots, U_N\}]
\]

where \( N \) is the number of child nodes of the priority selector.

**Parallel Node:** Since all child nodes of a parallel node \( j \) are meant to be executed simultaneously, the time interval is:

\[
[L_j, U_j] = [\max\{L_1, \ldots, L_N\}, \max\{U_1, \ldots, U_N\}]
\]

**Decorator:** The time interval of a decorator is set to the time interval of its child node.

### D. Emotional Selector

For every node in the EmoBT we can now calculate the risk, the time interval and the planning value. Next, it is needed to combine these values with the character’s current emotions in order to decide how to order the child nodes of the emotional selector. To mimic how affective states influence decision making in humans, as described in Section III, we introduce emotional weights for each of the three different factors: risk, time-discounting and planning.

Let \( r_1, \ldots, r_M \) (resp., \( r_1, \ldots, r_N \)) be the values of the emotions that positively (resp., negatively) affect the perception of risk. We define the emotional weight for risk as:

\[
E_{risk} = \frac{\sum_{i=1}^{M} r_i^+}{M} - \frac{\sum_{j=1}^{N} r_j^-}{N}
\]

Similarly, we define the emotional weight for time-discounting and planning:

\[
E_{time} = \frac{\sum_{i=1}^{Q} t_i^+}{O} - \frac{\sum_{j=1}^{P} t_j^-}{P}
\]

\[
E_{plan} = \frac{\sum_{i=1}^{R} p_i^+}{Q} - \frac{\sum_{j=1}^{R} p_j^-}{R}
\]
For each child node $i$ of the emotional selector, we calculate its weights based on the values of risk, planning and time. The weight for risk is calculated in the following way:

$$W_{\text{risk,}i} = (1 - E_{\text{risk}} \cdot \delta) \cdot \text{risk}_i$$

where $\text{risk}_i$ is the risk value for the child node $i$. Note that $W_{\text{risk,}i}$ should be clamped to the interval $[0, 1]$ since it represents a probability.

The weight for time is calculated in the following way:

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

The first part of the formula represents a hyperbolic time-discounting function used to describe human time-discounting [14], [15]. $\mu$ is a variable that is set to a value that fits the time span used in the simulation. $E_{\text{time}}$ is the emotional effect delay time calculated as:

$$\text{time} = L_i + \frac{U_i - L_i}{2}(1 - \sigma \cdot E_{\text{opt}})$$

where $E_{\text{opt}}$ is the emotional impact on optimization:

$$E_{\text{opt}} = \frac{\sum_{i=1}^{S} a_i^+ - \sum_{j=1}^{T} a_j^-}{S}$$

The weight for the planning is calculated as:

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

where $\omega$ is to fit the planning amount of the simulation. The variables $\delta, \lambda, \sigma$ and $\phi$ determine how much emotions affect the weights. Their value must be between 0 and 1, where 0 signifies no emotional impact and 1 corresponds to full emotional impact. The overall weight of the child node $i$ is calculated as:

$$W_i = \alpha \cdot W_{\text{risk,}i} + \beta \cdot W_{\text{time,}i} + \gamma \cdot W_{\text{plan,}i}$$

The constants $\alpha, \beta$ and $\gamma$ give importance to their respective factors.

To select which child node to execute, we list them in ascending order according to the weight value $W_i$. Hence, the lower value of $W_i$, the more desirable is the node. Since there are only three different factors to use to calculate the weights, but there may be many child nodes to an emotional selector, we want to make it possible, however unlikely, that the character will choose a less desirable node. To do so, after the child nodes have been ordered according to increasing weights $W_i$, we associate a probability $\text{prob}_i$ to every child node $i$:

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

where $a$ is chosen depending on which distribution one wants. Let us assume $a = 0.5$. This will make the most desirable child have a probability of 50% to be selected, the second child 25% and so on. Since there is always a small reminder left when one has calculated all probabilities for all nodes, that reminder is added to the last node to make sure an action is always chosen. Note that $a$ must be greater than 0.5, otherwise unwanted situations can happen. For instance, suppose we have three child nodes and we would have chosen $a = 0.2$. Then, the most desirable node would have had probability 20% to be chosen, the second child node 16%, and the last child node 64% due to the added reminder. While it is possible to make the probability of a node inversely proportional to the weight of the node, it may be the case that the weight values of different nodes will be fairly close in magnitude. This would in turn result in a rather equal probability distribution among the child nodes and there would hence be little noticeable difference in the character’s behavior.

Once the ordering has been established by selecting the child nodes based upon their probabilities, the emotional selector behaves as a priority selector. When the emotional selector has completed its execution, and it is executed again, the ordering of the nodes must be re-calculated.

**E. Example**

Consider the emotional selector given in Figure 2. It has three child nodes. The first one is a sequence selector $S1$ which has actions A1 to A7. All of these have a time-interval of $[5, 5]$ and zero risk. In other words, this sequence takes a long time to perform and involves quite a bit of planning, but has no risk. In the second sequence selector $S2$ there are 9 actions. The first action, B1, has a time-interval of $[0.1, 0.1]$ and a risk of 0.7. The rest of the actions take no time to perform and have no risk. This sequence is therefore fast but involves a lot of planning effort and a large risk. The third and final sequence $S3$ contains two actions, C1 with a time-interval of $[7, 7]$ and C2 with a time-interval of $[12, 12]$. The actions have the associated risks 0.7 and 0.8 respectively.

![Fig. 2. Emotional selector. The values for time, risk, and planning are listed below each child node.](image)
example, we let $E_{opt}$ be zero. The emotions mentioned here are derived from the psychology theories presented in Section III.

We let the specified emotions take different values to illustrate the effect this has on the action selection. In Table I, II and III we list the overall weights for the different factors given different emotional states. It can easily be seen that emotions greatly affect the factor weights in different ways, resulting in a different overall weight for the child nodes. When the character has a high level of fear, $S1$ is the most desirable choice, mostly because it involves no risk. When sadness is high, however, $S2$ is the most suitable option because it takes a much shorter time to execute. Finally, if fatigue has a high value, then $S3$ is selected since it consists of only two actions to be executed.

Suppose that the character has a high value of fear. Then, according to Table I, $S1$ is the most preferred node to be executed, $S2$ is the second best, and $S3$ is the least preferred. If we let $a = 0.7$, then the probabilities associated to $S1$, $S2$ and $S3$ are 70%, 21% and 9%, respectively. By randomly selecting the three child nodes with respect to the associated probabilities, we obtain the order of execution of the emotional selector.

### V. Case Study: NPC-fighter Behavior

We represent here an NPC-fighter behavior scenario with an emotional behavior tree. This example aims to illustrate the usefulness of our model in a typical game scenario. In the majority of games, NPCs have the need to fight an opponent, either the player or other NPCs. Often, the NPCs have different ways to attack an opponent, using different types of weapons, spells, and techniques. These different ways to attack the opponent have various benefits and drawbacks. While it is possible to hard-code the condition under which each technique should be used, this can cause the NPCs to be predictable, leaving them vulnerable to counter-attacks. Often, this can be circumvented by choosing an attack option randomly. This choice, however, leads to unpredictable behavior that cannot be logically explained from the gamer’s perspective. Instead, we wish to show how our emotional selector can be used to let the NPC choose the behavior which is the most suitable under the current emotional circumstances.

We design a simple fighting game scenario where the character has the following attack choices: it can throw grenades if it has any, it can use a musket or a sword, and it can do a fancy maneuver that includes a set of fast movements and knifing the opponent. The character should also escape when its health is too low and it should fall back to a patrolling behavior if there are no enemies nearby. The emotional behavior tree used for the case study is illustrated in Figure 3. In the

![Fig. 3. The behavior tree for the fighting game example.](image-url)
first child node contains a sequence of actions that tries to confuse the opponent so that the NPC can stab it. As described in Figure 4, each knifing action involves a risk of roughly 0.03, while the other actions contain no risk, making the total risk for the action sequence 0.1. The planning amount for the action sequence is 6. The movement actions have time-intervals [0.33, 1.0] while the knifing actions are considered instantaneous. The second child node represents throwing a grenade. The planning amount is small, 1, but the risk is high, 0.7, that the NPC will itself be injured in the attack. The time-interval for the action is [1.5, 2.5]. The third child node allows the NPC to attack the opponent using an old-time musket. These weapons take a long time to load, hence the action has a long time-interval of [7, 10] to complete. The risk, however, is relatively small 0.2 because there is no need for physically engaging with the enemy. The planning amount is set to 1.7 to account for the procedure of loading the weapon. The fourth and final child node describes a simple sword attack. The planning is minimal but the risk is large, 0.6, because it is a close-range attack. The time-interval for this sword attack is [2, 3].

For this scenario, we used the emotions fear, fatigue, sadness, anger and enjoyment. Constants $\alpha$, $\beta$ and $\gamma$ are set to 1.0 and constants $\mu$, $\lambda$, $\sigma$, $\delta$, $\omega$ and $\phi$ are set to 0.8, 0.9, 0.5, 0.9, 0.6 and 0.6 respectively. The results of this test can be seen in Figure 5 and 6. Figure 5 shows the weight values for each child node under different emotional influences. Figure 6 shows the corresponding probability distribution for $\alpha = 0.7$. For example, when the character is afraid, it wants to avoid risky actions, and hence prefers the fancy maneuver. Furthermore, when the character is sad, it chooses to attack with grenades because it wants to have quick results, and ignores the risks involved. This behavior is consistent with theory on how emotions affect human decision making, as described in Section III. It is worth noticing that the weight values for the different child nodes are fairly similar. Should one change some of the factors of the child nodes, the outcome may be different. However, since there is a probability that a slightly less desirable behavior will be chosen over a more desirable one, the fairness of the selection mechanism is guaranteed, that is, a behavior will be eventually selected for execution.

VI. CONCLUSION

In the past few years, behavior trees have become increasingly popular within the gaming industry. Behavior trees, however, lack the possibility to mimic the dynamic emotional choices that are so important for realistic behavior. In this paper we have presented a model for behavior trees that introduces a new type of selector, the emotional selector. This selector incorporates the character’s emotional state into the decision making, letting the emotions affect the character’s view on risk, time-discounting, and planning. Furthermore, we have demonstrated the usefulness of our model using a case study.

The proposed model is fully implemented. It is fast since most of the calculations only have to be done once (calculating the derived planning/risk/time values from child nodes) and scales well ($O(n)$) with the size of the behavior tree. One of the benefits of our approach is that the game designer is in control of which parts of the tree will use the emotional selection. While it is possible to replace all priority selectors with emotional selectors, this is inadvisable. To maintain
control, we suggest emotional selectors be used where the child nodes represent a set of actions that perform behaviors in the same category, such as different ways of attacking or different ways of looking for resources. No change is made to the behavior of the original nodes of the tree. These can be used as before.

Choosing which emotions should affect what factors is left to the designer. Despite a substantial amount of scientific literature concerning emotional impact on decision making, it is currently impossible to define exactly which emotions affect each factor and how much. This is only natural, as measuring human emotions is still extremely difficult, at best. Instead, one must choose emotions that give an overall behavior that is consistent with a thinking and feeling character.

Other attempts have been made at creating emotional decision-making systems [1], [11], [29]. None of these let emotions affect such factors as risk, planning or time. Instead, the focus is on either coping mechanisms [1], [29] or the narrowing of the agent’s focus of attention [11].

While our case study presents the theoretical usefulness of our model, it is clear that more tests are needed to show the actual benefits. Since believability of NPCs is our main goal, the model should be tested through user studies where gamers are exposed to NPCs using the suggested model. To do so, however, the characters must be able to display their emotions in an intuitive way (through body postures, facial animations, etc.). This is currently not possible in our architecture, which still under development. We therefore leave the user tests as future work and instead propose our model as a way towards extending behavior trees with emotions.

REFERENCES


