A computational experiment involving decision-making techniques

Eduardo F. S. de Almeida André R. da Cruz*

Centro Federal de Educação Tecnológica de Minas Gerais, Departamento de Computação e Construção Civil, Brazil

ABSTRACT

This work performs an experimental study involving three artificial intelligence techniques for decision-making and two character models. A platform shooting game environment was implemented, in two dimensions, in order to be compared the techniques based on decision tree, state machine and Markov chain. For this purpose, it was planned and executed a computational experiment in which the player faces, individually and in random order, each combination of character and technique. For each player, samples are stored with information about fun, difficulty and time spent to defeat each agent. The experiment results are analyzed using the nonparametric Quade test with confidence 0.95. In the end, a comparative statistical analysis is presented about which combinations of decision technique and character model are significantly more fun, difficult and requiring more time to defeat them.

Keywords: Decision-making models, Decision trees, State machine, Markov chain, Artificial intelligence for games.

1 INTRODUCTION

According to Rabin [6], the search for digital games is in constant increasing. This high demand and with the improvement of existing technologies, people have become more stringent and want them to be more complex than before.

The use of Artificial Intelligence (AI) techniques in games proposes to extend interactivity and gameplay. Depending on the application, as affirmed by Millington and Funge [4], specific algorithms are used to simulate a behavior pattern such that agents appear with reality. Particular examples of applied works can be found in texts wrote by Cunha and Chaimowicz [1], and Silva, Silva, and Chaimowicz [9].

An assessment in relation to the subjective quality of an AI technique is of great importance for future decisions in a development team. It is desired to create characters with high acceptance in relation to different kind of players according to some features. In this way, this paper proposes a methodology in order to compare agent models in a computational experiment environment.

The particular case study involves three simple artificial intelligence techniques for Decision-Making (DM) and two character models, with a finite set of behaviors, in a two dimensions platform shooting game environment. The techniques are based on Decision Tree (DT), State Machine (SM) and Markov Chain (MC) [4].

In the planned computational experiment, the player faces, individually and in random order, each combination of character and technique. For each player, samples are stored with information about fun, difficulty and time spent to defeat each agent. The null hypothesis about the equality of these measures will be tested with the nonparametric Quade test (see Garcia et al. [2] for more details) with confidence 0.95.

The comparative statistical analysis will present which combinations of decision technique and character model are significantly more fun, difficult and requiring more time to defeat them.

*e-mail: andrecruz@timoteo.cefetmg.br

The main contribution of this work is the presentation of a methodology capable of comparing, in an environment of uncertainty, a set of agents modeled with different techniques. Clearly, the method proposed here, without loss of generality, is extensible to compare any number of combinations of techniques applied to digital games.

This paper is organized in the following way. Section 2 presents briefly the theoretical framework. Section 3 explains the methodologies and the experimental design. Section 4 provides the results. Finally, Section 5 concludes the paper.

2 THEORETICAL FRAMEWORK

This section provides some concepts used in this work.

2.1 Artificial Intelligence and Decision-Making

Russell and Norving declare that AI is an area of knowledge that has a set of techniques that aims to reproduce apparently intelligent behaviors in virtual or real agents. This includes communication, learning, and decision-making [8]. Designers aim to develop games that present interesting challenges with smart opponents. For this purpose, a DM algorithm, that determines what to do in different situations, can be applied in a character modeling. It is common in games with simpler agents, the application of DM systems based on DT, SM or MC [4]. However, it is not intuitive which of these techniques can generate a better immersion associated with a model.

DM is the component that is linked to the actions, implemented in agents, which allows them to assume behaviors. Although there are several different techniques, all have the same general structure. The agent observes the environment, processes a group of information and decides on an action to perform. The data is the knowledge that a certain character gets into the environment, and the result is an action. Such knowledge can be divided into two, internal and external. The latter is the information absorbed from the environment, such as the position of the enemy, the weapon's ally, the level of health and the direction in which comes a sound. And internal knowledge is the information that the character has about his own, such as their own health level, the set of movements performed previously and the priority objectives [4].

The format and amount of knowledge depend on the game requirements. The representation is essentially linked with the majority of DM techniques [4].

Following, it is presented the three techniques of this work.

2.2 Decision Tree

A Decision Tree, DT, is a set of chained rules like "IF–ELSE", which can be represented by a tree. A DT has as input an information described by a set of features, and as output an action or decision. It represents a function that takes a vector of attributes, performs a sequence of tests and returns an output value [8].

A binary DT has a root, nodes and leaves. Each node corresponds to a condition or an output, and the edges are considered the possible results of a condition. For each conditional node, the agent must decide for an option, generally true or false, based on the game global knowledge which is programmed previously. The root node is the first decision which must be done. Then a sequence of choices is selected until achieve the end of the tree, which is a leaf with a final output decision [4].

When a DT is applied in a character model, each choice is performed based on the model determined by a set of rules defined previously. A DT is here implemented deterministically, which means that the set of rules happens always in same way. In general, agents controlled by DTs are computationally simple, fast, and act based on the game's global state [4].

The main advantage of a DT is its simplicity to understand, model and implement. A disadvantage is the lack of randomness which allows players preview future actions.

2.3 State Machine

The study of State Machine, SM, belongs to automata theory. A SM can model a system from its finite set of behaviors and their related transitions [7]). It is the technique most often used for DM in games [4].

A finite state machine $M = (S, I, O, f, g, s_0)$ consists of a finite set S of states, a finite input alphabet I, a finite output alphabet O, a transition function f that assigns to each state and input pair a new state, an output function g that assigns to each state and input pair an output, and an initial state s_0 [7].

A SM can be represented graphically by a state diagram in a graph format. Each node will denote a different state and each edge a possible transitions given a stimulus. Also, there is an indication of the initial state [7].

This kind of technique is easily applied in agent modeling if the same is considered as a SM. For a character, a possible set of behaviors can be considered as set of states and the respective changes can be defined as the transitions [4].

SM has a merit of taking account of both the world around them and their internal makeup. A disadvantage is that the speed up over this approach is quite small, unless the scripting language includes some kind of compilation into machine code.

2.4 Markov Chain

A stochastic process is defined to be an indexed collection of random variables $\{X_t\}$. The index t runs through a given set S of states, and X_t represents a unique measurable characteristic of interest at discrete time indexed by t [3].

A stochastic process $\{X_t\}$ is said to have the Markovian property if $P(X_{t+1} = j | X_0 = k_0, X_1 = k_1, \ldots, X_{t-1} = k_{t-1}, X_t = i) = P(X_{t+1} = j | X_t = i)$, for $t = 0, 1, \ldots$ and every sequence $i, j, k_0, k_1, \ldots, k_{t-1}$. The conditional nonnegative probability $P(X_{t+1} = j | X_t = i) = P_{ij}$ is called transition probability. A stochastic process is a Markov Chain, MC, if it has the Markovian property [3].

MC is of interest here for describing the random behavior evolution of a game agent over some period of time. Given n possible states for character, a convenient way of showing all probabilities is through the transition matrix. For a fixed state i, the i-th row shows all transition probabilities for any other j state. As each row represents a probability distribution, it is true that $\sum_{j=1}^{n} P_{ij} = 1$ [3].

Unlike the other two techniques, MC implements randomness in the behavior of agents. On the other hand, it may be complicated to determine a satisfactory probability distribution, for each state.

2.5 Statistical Hypothesis Testing

A statistical hypothesis is a statement about the parameters of one or more populations. The affirmation is called hypothesis, and the DM procedure it is called hypothesis testing [5].

Statistical hypothesis testing and confidence interval estimation of parameters are the fundamental methods used at the data analysis stage of a comparative experiment. A procedure leading to a decision about a particular null hypothesis against an alternative is called a test of a hypothesis. The null hypothesis is the one desired to be tested. Rejection of the null hypothesis leads to accepting the alternative hypothesis [5].

	Treatment			
Block	**1**	**2**	...	**K**
1	x_{11}	x_{12}	...	x_{1K}
2	x_{21}	x_{22}	...	x_{2K}
⋮	⋮	⋮	⋮	⋮
B	x_{B1}	x_{B2}	...	x_{BK}

Table 1: Samples arranged in a table for nonparametric Quade test.

Testing the hypothesis involves taking random samples, computing a test statistic from the sample data, and then using an appropriate test statistic to make a decision about the null hypothesis [5].

2.6 Quade Test

The nonparametric Quade test analyzes $K \geq 2$ treatments in a randomized complete block design experiment with $B > 1$ blocks [2]. Samples are organized in a table, in which each row represents an independent block and each column a treatment (see Table 1).

The test computes weighted ranks for each treatment in the following way. Let $R(x_{ij})$ be the rank assigned to x_{ij} within block i. Average ranks are used in the case of ties. Compute the range of each block i and assign in Q_i. Let $S_{ij} = Q_i(R(x_{ij}) - (K+1)/2)$ and $S_j = \sum_{i=1}^{B} S_{ij}$, for all $j = 1, 2, \ldots, K$.

Following, it is considered the null hypothesis which affirms that all treatments have identical effects, against the alternative, which declares that at least one treatment is different. The test statistic used is $T = (B-1)\beta/(A2 - \beta)$ in which $A2 = \sum_{i=1}^{B}\sum_{j=1}^{K} S_{ij}^2$ and $\beta = (1/B)\sum_{j=1}^{K} S_j^2$.

Given a significance level α, the null hypothesis is rejected if $T > F_{(\alpha, K-1, (B-1)(K-1))}$, in which F is the percent point function of the F distribution. In this case, it is determined the different pairs of treatments using Equation 1, in which t comes from t-Student distribution.

$$|S_i - S_j| > t_{(1-\alpha/2,(B-1)(K-1))} \sqrt{\frac{2B(A2-\beta)}{(B-1)(K-1)}} \qquad (1)$$

3 Applied Methods

It is described here the characters models, the DM techniques developed to guide the enemies and the experimental design.

3.1 Characters

It was modeled a player character (PC) and two enemies (P1 and P2) controlled by one of the three DM techniques based on DT, SM and MC. The three models have the same moves (walk and jump), but they have different ways to attack.

PC can attack with a revolver, machine gun, shotgun or grenades. The P1's weapons are revolver, knives and rockets. P2 can use a bazooka, machine gun, knives or grenade of German farting to attack.

All enemy movements and attacks are controlled by states, which are defined according to the set of actions that a character can perform. Each state has its own peculiarities, such as the type of movement to be held and the required time that should stay in it.

The set of states for P1 are "*On guard*", "*Pursuing*", "*Attacking 1*" (revolver), "*Attacking 2*" (knives), "*Attacking 3*" (rockets), "*Defending*" and "*Retreating*". P2 has the states "*On guard*", "*Pursuing*", "*Attacking 1*" (bazooka), "*Attacking 2*" (machine gun), "*Attacking 3*" (knives), "*Attacking 4*" (grenade of German farting), "*Defending*" and "*Retreating*". The time required in each state is not shown here because of space limitations. The transitions among states are controlled by a DM technique.

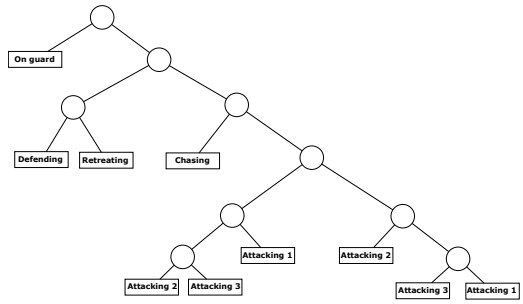


Figure 1: DT for P1 with questions in nodes and states in the leafs.

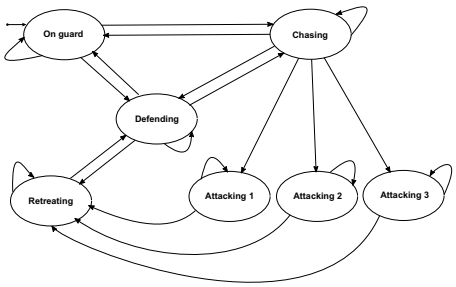


Figure 2: State diagram for P1 with all possible transitions.

3.2 Decision-Making Models

It was modeled and applied for each enemy, the DM techniques presented in Sections 2.2, 2.3 and 2.4. Therefore, there are six enemy-technique combinations (avatars), which are known as P1-DT, P1-SM, P1-MC, P2-DT, P2-SM and P2-MC.

A binary DT was created for P1 and P2, with Boolean questions in relation to the environment in order to select the next state. Figure 1 illustrates a simplified DT for P1.

In order to change its state, an enemy with SM has a transition function which uses as parameters its current state, position, dimensions, and life level, jointly with the player's life level. If the environment determines that the conditions for a transition are fulfilled, then there is the change of the current state to the target state. Figure 2 presents the state diagram for P1.

The MC procedure is simple. When the character is in a state, after the last residence time, it is raffled a random number that defines the next state. The avatars with SM and MC share the same states and transition possibilities, as presented in Figure 2, with the difference that the second has a stochastic behavior.

In general, in an adequate period of time, the transition between two states is computed from a set of input parameters. The DM technique analyzes this input and selects the next state according to particular rules. This process is repeated while the agent is active.

3.3 Experimental Design

In order to compare and analyze the behaviors in the categories of fun, difficulty and time taken to be defeated, it was built an environment simulating a 2D platform shooting game. Each independent and complete block of samples are obtained in a session, when a player faces all six avatars, represented by rectangles, in random order. In this blind experiment, it is not known which combination of character and DM technique is faced.

In each confrontation of a session, the player begins with five lives and 100 health points. In each life, the revolver has unlimited ammunition, 200 shots for machine gun, 20 shots for shotgun and 5 grenades.

When an avatar is defeated, two questions must be answered using a slider bar with continuous range in $[0, 1]$. The position value of the slider bar will be used to measure the feeling of the players. The first question is "*What was the difficulty to defeat the avatar?*". The more on the left the slider is, the easier it was the feeling of defeating the avatar. On the contrary, the more on the right, the harder it was to defeat the enemy. The other question is "*What level of fun did you have when you confronted this last avatar?*". The more on the left the slider is, the boring it was, and the more the on right, the funnier it was. In addition to these questions, it is internally computed the time in seconds the player took to defeat the avatar. These data are stored for each confrontation and used to analyze the avatars posteriorly.

Three null hypothesis will be tested with significance level $\alpha = 0.05$ using the nonparametric Quade test. The first affirms that *all avatars have the same difficulty to be defeated*. The second declares that *all avatars generate the same level of fun*. And the last expresses that *all avatars require the same time to be defeated*. These experimental results will be presented in the next section.

4 Results and Analysis

The application was available on-line during two months and it was possible to collect 142 records for each avatar in each question. The players were invited from Facebook communities related to game enthusiasts, game development and programming. They were advised to play only once in order to not bias the data.

The box plots for all collected data are presented in Figure 3. It is possible to observe the data dispersion for difficulty, fun and time spent to defeat the avatars. The Quade test outputs for the three hypothesis tests are presented in Figure 4. For each avatar, it is shown the weighted ranking with its respective critical interval. Intervals without intersection present significant differences in accordance to Equation 1. Otherwise, there is not an evidence of statistical difference.

In relation to difficulty, fun and time spent to defeat an enemy, Quade test returned respectively the p-values $4.10 \cdot 10^{-6}$, $5.16 \cdot 10^{-4}$ and $1.20 \cdot 10^{-5}$. This indicates that at least one avatar differs from another in all analyzed features. Therefore, firstly, it can be concluded that an adequate combination of a character model and a DM technique is important in game design to select or discard avatars in a project according to some quality criteria.

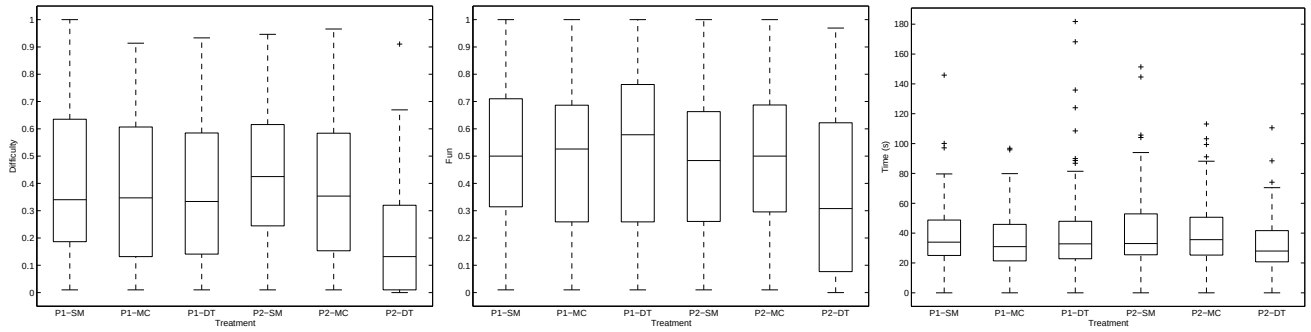
In general, Figures 3 and 4 clearly indicate that the combination of P2 with DT was the easiest and least fun avatar with the smallest time to defeat it. Because of these low level of qualities, this model is one to be removed from the project.

In relation to difficulty, the others avatars were competitive and did not presented statistical difference. Although P1-DT is in this not differentiated group, it can be observed that the same is located more on the left in Figure 4a. This is an indication that DM based on DT can be easier than others. It can be also noted that P1-SM, P2-SM and P2-MC have a big and very similar weighted rankings, indicating a high similarity in difficulty. A designer must consider to insert one of these in a game as a harder avatar.

According to fun, with the exception of P2-DT, the avatars did not statistically differentiate. A highlight is given to P1-DT that although it is on the left in its group of difficulty in Figure 4a, the same is more on the right, indicating a high acceptance of fun in Figures 3b and 4b. This avatar would be interesting to be inserted in initial phases, once it is very fun and not too difficult.

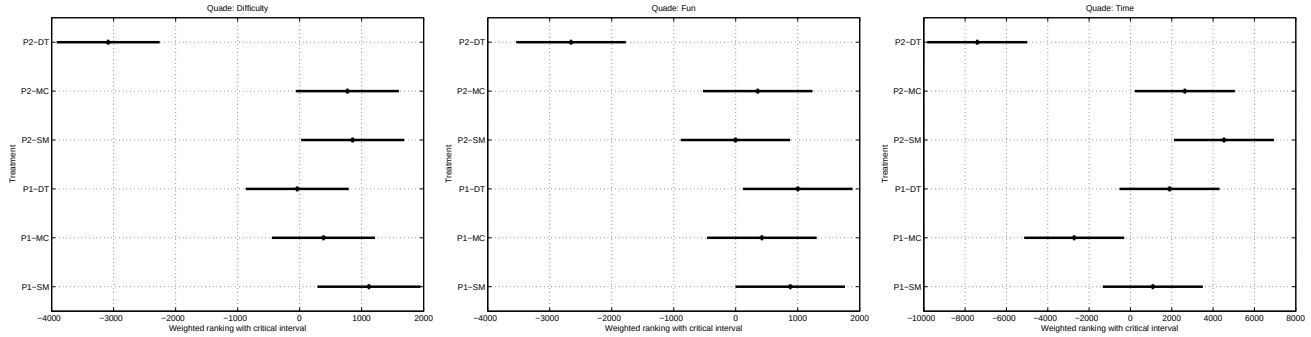
Another attractive avatar is P1-SM, which presented a good difficulty, a high level of fun and an intermediate time spent to defeat it. For these facts, this character is a good indication of a middle challenge.

At analyzing in detail the time spent to defeat the avatars in Figure 4c, it can be concluded that P2-SM is one which presented more



(a) Difficulty (b) Fun (c) Time in seconds

Figure 3: Boxplot output for difficulty, fun and time to defeat the enemy.



(a) Difficulty (b) Fun (c) Time

Figure 4: Quade test output for difficulty, fun and time to defeat the enemy.

time. In addition, the same is one of hardest to be defeated and not so fun. Therefore, it is an interesting candidate of a final challenge.

5 Conclusion

This paper presented an experimental methodology in order to compare avatars, according to three quality criteria – difficulty, fun and time spent to defeat an avatar – considering the character model and DM techniques. It was created a simulation of a 2D platform shooting game environment in JavaScript and HTML5.

The application was available on-line and players faced against six avatars, once and in random order. Each avatar was a combination of a character model and one of three decision-model techniques, which are Decision Tree, State Machine and Markov Chain. After winning each enemy, the player answered a quiz in respect to difficulty and fun. It was also computed the duel time.

After two months, the stored data of this complete block design experiment were collected and analyzed using the nonparametric Quade test. It was concluded that a combination of character model and DM technique can statistically differentiate in some subjective quality criteria according to players. It also can be observed which avatar is better suited during the evolution of a game, according to its difficulty, fun and time spent to defeat it.

This methodology is an interesting action to be applied during the game development phase from alpha to beta. Once it is possible to measure and statistically analyze any uncertain quality criteria of avatars, the final product tends to be more organized in terms of gameplay and immersion.

It can be also noted that the methodology presented here is easily extended to analyze avatars with different techniques.

For future works some actions are intentioned. More decision-making techniques will be included in this research, like Fuzzy Logic, Goal-Oriented Behavior and Rule-Based Systems [4]. It is desired to obtain a bigger number of samples to improve data representativeness. It is intended to apply other statistical tests to compare avatars using other different quality criteria and to analyze interactions among factors. It is also interesting to collect samples from different groups of players in order to analyze profiles and to create games directed to them.

References

[1] R. L. F. Cunha and L. Chaimowicz. An artificial intelligence system to help the player of real-time strategy games. In *SBC - Proceedings of SBGames 2010*, Florianópolis, Brazil, November 2010.

[2] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044 – 2064, 2010. Special Issue on Intelligent Distributed Information Systems.

[3] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw Hill, 9 edition, 2010.

[4] I. Millington and J. Funge. *Artificial intelligence for games*. CRC Press, 2 edition, 2009.

[5] D. C. Montgomery and G. C. Runger. *Applied statistics and probability for engineers*. Wiley, 5 edition, 2010.

[6] S. Rabin. *Introduction to Game Development*, volume 2. Charles River Media, 2009.

[7] K. H. Rosen. *Introduction to Discrete Mathematic and Its Applications*. McGraw Hill, 6 edition, 2007.

[8] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 3 edition, 2009.

[9] M. P. Silva, V. N. Silva, and L. Chaimowicz. Dynamic difficulty adjustment through an adaptive ai. In *SBC - Proceedings of SBGames 2015*, Teresina, Brazil, November 2015.